STANFORD UNIVERSITY

CENTER FOR SYSTEMS RESEARCH

Synthesis and Analysis of Adaptive Array Processors

by

Bernard Widrow
Otis Lamont Frost, III
James Edward Brown, III

Information Systems Laboratory

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
ADAPTIVE SYSTEMS ADAPTIVE ARRAYS CONSTRAINED LEAST-SQUARES ALGORITHM MINIMUM MEAN-SQUARE ERROR ADAPTIVE FILTER KALMAN FILTER						

Part II Abstract contd.

The purpose of this research is to develop and analyze a gradient-descent surface-searching algorithm for automatically adjusting (adapting) the parameters of a linear tapped-delay-line array processor in order to improve its performance in an unknown changing environment. The tracking ability of this algorithm is demonstrated when the characteristics of the nonstationarity are such that the optimum parameter sequence can be modeled as a first-order Markov process with a known transition function. A worst-case analysis of the algorithm is presented for three types of nonstationarities when the above model for the nonstationarity is not applicable.

The techniques developed in analyzing the above algorithm provide a powerful approach for the further study of gradient-descent algorithms used in searching unknown, nonstationary surfaces. Among the most consequences are:

i) the removal of the usual assumption that the data be jointly Gaussian;

ii) the development of a new convergence theorem for a dynamic stochastic approximation algorithm, thereby extending a branch of stochastic approximation theory to the analysis of adaptive processors in nonstationary statistics;

iii) the enlargement of the class of problems for which stochastic approximation algorithms, adaptive estimation algorithms, and the Kalman-Bucy theory can be compared.

Also presented in an appendix is a procedure for automatically adjusting the convergence factor. Some experimental results are presented.

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Electronics Laboratories Stanford University, Stanford, Calif.	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP NA

3. REPORT TITLE
SYNTHESIS AND ANALYSIS OF ADAPTIVE ARRAY PROCESSORS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
FINAL REPORT on Contract (6-27-69 to 4-30-70)

5. AUTHOR(S) (First name, middle initial, last name)
Bernard Widrow, Otis Lamont Frost, III and James Edward Brown, III

6. REPORT DATE January 15, 1971	7a. TOTAL NO. OF PAGES 234	7b. NO. OF REFS 91
8a. CONTRACT OR GRANT NO. N00024-69-C-1430 b. PROJECT NO c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) SU-SEL-71-004 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10. DISTRIBUTION STATEMENT
This document has been approved for public release and sale. Its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY NAVSHIPS, Dept of the Navy, Washington, D.C.

13. ABSTRACT

PART I

The problem considered in this report is to find the vector of weights W minimizing $E\{[d(t) - W^T X(t)]^2\}$ subject to linear equality constraints on W, where X(t) is a vector of random variables measured at time t and d(t) is a random variable related to X(t). This is a classical problem in linear estimation theory, except that the statistics of the random variables are assumed unknown and must be learned through observations. A computationally simple procedure, called the Constrained Least-Mean-Squares algorithm, is proposed for processing the observations and is shown to converge to the optimal linear processor. The algorithm is useful in real-time modeling, filtering, and estimation, particularly in cases where the optimal time-varying linear processor (e.g., Kalman filter) cannot be used because of computational complexity or lack of necessary information about the system. Special attention is given to real-time processing of data from an array of sensors, and it is shown that the Constrained Least-Mean-Squares algorithm permits implementation of an array processor that requires very little a priori statistical information.

PART II

In the classical design of processors for sensor arrays whose purpose is signal detection and estimation, a receiver is optimized on the basis of the a priori knowledge of the statistics of its input signals. However, when the a priori knowledge is not available, the receiver's performance can still be improved by performing measurements on its input signals and incorporating this new information into its design. Such receivers are called adaptive. (contd. on back)

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801

Unclassified
Security Classification

SEL-71-004

SYNTHESIS AND ANALYSIS OF ADAPTIVE ARRAY PROCESSORS

by

Bernard Widrow
Otis Lamont Frost, III
James Edward Brown, III

15 January 1971

FINAL REPORT

Prepared under

NAVSHIPS, Department of the Navy
Contract N00024-69-C-1430

Information Systems Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

FOREWORD

This is a Final Report under Contract N00024-69-C-1430, covering a period of research from 27 June 1969 to 30 April 1970. The report consists of two parts:

Part I, by O. L. Frost, III, is based on his Ph.D. thesis. The principal contribution is a new adaptive algorithm for minimizing mean-square-error in an adaptive processor which simultaneously subjects the weight-vector components to a linear equality constraint. When applied to adaptive arrays, this algorithm allows one to obtain precise control of the array frequency response and gain level in the "look direction" while minimizing mean-square-error. Frost's algorithm is probably the best yet devised for adaptive arrays.

Part II, by James Edward Brown, III, is based on his Ph.D. thesis. This work is highly theoretical and presents a framework for mathematical analysis of adaptive processors when subjected to changing (non-stationary) signal and noise fields. For various adaptive algorithms, rate of convergence and variance of the weight vectors are analyzed. This work is general, and applicable to a wide variety of adaptive signal processors.

PART II

by

James Edward Brown, III

PART I.

ADAPTIVE LEAST SQUARES OPTIMIZATION SUBJECT TO LINEAR EQUALITY CONSTRAINTS

by

Otis Lamont Frost, III

ABSTRACT

The problem considered in this report is to find the vector of weights W minimizing $E([d(t) - W^T X(t)]^2)$ subject to linear equality constraints on W, where $X(t)$ is a vector of random variables measured at time t and $d(t)$ is a random variable related to $X(t)$. This is a classical problem in linear estimation theory, except that the statistics of the random variables are assumed unknown and must be learned through observations. A computationally simple procedure, called the Constrained Least-Mean-Squares algorithm, is proposed for processing the observations and is shown to converge to the optimal linear processor.

The algorithm is useful in real-time modeling, filtering, and estimation, particularly in cases where the optimal time-varying linear processor (e.g., Kalman filter) cannot be used because of computational complexity or lack of necessary information about the system. Special attention is given to real-time processing of data from an array of sensors, and it is shown that the Constrained Least-Mean-Squares algorithm permits implementation of an array processor that requires very little a priori statistical information.

TABLE OF CONTENTS (PART I)

ACKNOWLEDGMENT

The author wishes to thank Dr. Bernard Widrow for his guidance, encouragement, and support during this research. Sincere appreciation is given to Dr. Michael Arbib for many stimulating discussions, and to Dr. William Spicer for his helpful comments and suggestions on the presentation of the material.

The author particularly wishes to thank his friends and colleagues at Stanford Electronics Laboratories, Jim Brown, Michel Installe, John Moschner, Lloyd Griffiths, Tom Daniell, Barbara Kenyon, Burwell Goode and many others, for their assistance.

Special thanks go to Judith Ann for her considerable role in the course of this research.

I. INTRODUCTION

This paper presents a simple algorithm for minimizing a quadratic cost criterion subject to linear equality constraints. The technique, called the "Constrained-Least-Mean Squares" or "Constrained LMS" algorithm is an iterative, stochastic gradient-descent algorithm with low memory requirements. Computationally, it is simple enough that for a variety of practical problems it can be implemented in real time on a small general-purpose computer.

The algorithm is applicable to problems in least squares filtering, estimation, modeling, and others which may properly be viewed as linear-constrained quadratic optimization problems. Specific examples treated in the paper include real-time minimum-variance unbiased estimation, consistent modeling that includes known linear constraints on the model parameters, and real-time processing of data from an array of antennas or other sensors. The constrained least-mean-squares approach is particularly interesting in the estimation and array processing applications because it requires very little a priori information for implementation.

The rate of convergence of the algorithm is studied and its steady-state performance is compared with the optimum. A gain constant is shown to control a tradeoff between fastest convergence rate and best steady-state performance. By suitable choice of gain the steady-state performance of the algorithm can be made arbitrarily close to the performance

of the optimum least-squares filter.

Previous work on unconstrained least-squares array processing was done by Griffiths [12]; his method requires knowledge of second-order signal statistics. Widrow, et al. [30] proposed a variable-criterion optimization procedure involving the use of a known training signal; this was a direct application of the original work on adaptive filters done by Widrow and Hoff [29]. Griffiths also proposed a constrained least-mean-squares processor not requiring a priori knowledge of the signal statistics [11]; a new derivation of this processor, given in Appendix A, shows that it may be considered as putting "soft" constraints on the processor via the quadratic penalty function method.

"Hard" (i.e., exactly)-constrained iterative optimization was studied by Rosen [23] for the deterministic case. Lacoss [14] and Booker [1] studied "hard"-constrained stochastic optimization in the array processing context. All three authors used "gradient projection" techniques; Rosen and Booker correctly indicate that gradient projection methods are susceptible to cumulative roundoff errors and are not suitable for long runs without an additional error-correction procedure. The Constrained LMS algorithm is designed to avoid error accumulation while maintaing a "hard" constraint; as a result, it is able to operate continually in order to track an environment that may be slowly time-varying. Discussion of gradient-projection methods and

a comparison of the error-correcting properties of the two algorithms is given in Section VII.

In the following section, the general constrained least-mean-squares problem is formulated as a theorem and the optimal solution is derived under the assumption that all the relevant statistics of the problem are known. Several corollaries applying to interesting special cases are drawn. The optimal solution is seen to be computationally difficult, requiring a number of matrix multiplications and inversions. In Section III, the computationally simple Constrained LMS algorithm is derived that converges to the optimal solution while learning the statistics of the problem. This algorithm and studies of its properties is the principal result of this thesis. Special forms of the general algorithm are used to solve particular problems. Remaining sections are concerned with geometrical interpretation of the algorithm, its performance, applications, and computer simulations.

II. CONSTRAINED LEAST-MEAN-SQUARES OPTIMIZATION

A. Notation

In this paper a vector is taken to be a column vector. The superscript T denotes transpose. The expected value of a quantity $\{Q\}$ is denoted by $E\{Q\}$ or $\overline{Q}$. The matrix of correlations between two vectors of random variables, A and B, is written $E\{AB^T\} = R_{AB}$; the vector of correlations between a vector X and a scalar d is written $E\{Xd\} = R_{Xd}$. A vector of zeros of arbitrary dimension is θ and the matrix of zeros is $\underline{\underline{0}}$

B. The General Problem and Optimal Solution

There are two purposes for this section. The first is to define the general constrained least-mean-squares problem and derive the optimal solution. This solution could be obtained directly if one knew the problem statistics beforehand. It will be shown later that the Constrained LMS algorithm converges to this solution and can be used when the problem statistics are unknown. The second purpose is to show that several interesting and important problems can be put in the framework of the general constrained LMS problem and therefore are solvable by the algorithm.

Let X be a vector of n observed data points, $X^T = (x_1, x_2, \ldots, x_n)$, that are drawn from a distribution with $E\{XX^T\} = R_{XX}$. Let d be a random variable correlated with X by an n-dimensional correlation vector R_{Xd}. In this section

R_{XX} and R_{Xd} are assumed known. Let W be an n-dimensional vector of weightings that will be applied to X to estimate d. Let the estimate of d be

$$y \triangleq W^T X , \tag{2.1}$$

and the error between d and the estimate be

$$e \triangleq d-y . \tag{2.2}$$

The constrained least-mean-squares optimization problem is to find the weight vector W_* that minimizes the expected squared error in the estimate,

$$E\{e^2\} = E\{[d-y]^2\} = E[d - W^T X]^2\} \tag{2.3}$$

subject to certain linear equality constraints on W.

The reason for placing constraints on W was suggested in the introduction and will be made clear in the applications. In general, m linear equality constraints (with $n > m$) are placed on W of the form

$$c_i^T W = f_i , \quad i = 1,2,\ldots,m , \tag{2.4}$$

where each c_i is an n-dimensional vector and each f_i is a scalar constant. This is a set of m simultaneous equations which the n components of W must satisfy, but since $m < n$, the equations do not completely determine or totally constrain W. Therefore W can be optimized, to minimize a mean square error, subect to the linear constraint (2.4). It is well known that by requiring W to satisfy $c_i^T W = f_i$ for any single i restricts W to lie in an (n-1)-dimensional hyperplane. Similarly, it is shown in Section IV that constraining

W to satisfy the m equations of (2.4) restricts W to an $(n-m)$-dimensional plane† if the vectors c_i are linearly independent. To express the constraints in matrix notation define

$$C \triangleq \overbrace{\begin{bmatrix} c_1 & c_2 & \cdots & c_m \end{bmatrix}}^{m} \Big\} n \;, \quad \mathcal{f} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} . \tag{2.5}$$

The constraint matrix C is $(n \times m)$ with $n > m$. It will be assumed that the constraint vectors c_i are linearly independent so that by the definition of rank as the number of linearly independent columns of a matrix, C has full rank equal to m. The constraints (2.4) are now written

$$C^T W = \mathcal{f} \;. \tag{2.6}$$

The problem is summarized and the solution is given in the form of a theorem.

Theorem 1. (Constrained Linear Least-Mean-Squares Optimization) Let d be a random variable and X be an n-dimensional vector of random variables with known correlation matrices

†Other names for an "r-dimensional plane" are "linear variety" and "Linear manifold".

$$E\{XX^T\} = R_{XX} \qquad (n \times n)$$

$$E\{Xd\} = R_{Xd} \qquad (n \times 1)$$

and R_{XX} positive-definite. The optimum constrained least-mean-squares weight vector solving

$$\begin{aligned} &\text{minimize} \quad E\{[d - W^T X]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \end{aligned} \tag{2.7}$$

where C is an $(n \times m)$ matrix $(n > m)$ of full rank and $\mathcal{F}$ is an m-vector, is

$$W_* = [I - R_{XX}^{-1} C (C^T R_{XX}^{-1} C)^{-1} C^T] R_{XX}^{-1} R_{Xd} + R_{XX}^{-1} C (C^T R_{XX}^{-1} C)^{-1} \mathcal{F} . \tag{2.8}$$

The optimum constrained linear least-mean-squares estimate of d is $y = W_*^T X$.

Proof of Theorem 1.

The proof uses the method of Lagrange multipliers, which is basic to the later development of the major algorithm and another proof. A geometrical interpretation of Lagrange multipliers expressed in the context of this work is presented in Appendix E.

The cost function is $J(W) = E\{[d - W^T X]^2\}$

$$= E\{d^2\} - 2\, E\{W^T X d\} + E\{W^T X X^T W\}$$

$$= E\{d^2\} - 2 W^T R_{Xd} + W^T R_{XX} W \; . \tag{2.9}$$

Including a factor of ½ to simplify later arithmetic, adjoin the constraint function to the cost function by a m-dimensional vector of undetermined Lagrange multipliers λ:

$$H(W) = \tfrac{1}{2}J(W) + \lambda^T(C^TW - \mathcal{F})$$

$$= \tfrac{1}{2}[Ed^2 - 2W^TR_{Xd} + W^TR_{XX}W] + \lambda^T(C^TW - \mathcal{F}) \ . \quad (2.10)$$

The necessary conditions for optimality are

$$\nabla_W H(W) = \theta \ , \quad (2.11)$$

and

$$C^TW = \mathcal{F} \ . \quad (2.12)$$

Taking the gradient of (2.10) with respect to W

$$\nabla_W H(W) = -R_{Xd} + R_{XX}W_* + C\lambda = \theta \quad (2.13)$$

and solving for the optimal weight vector

$$W_* = R_{XX}^{-1}R_{Xd} - R_{XX}^{-1}C\lambda \ , \quad (2.14)$$

where R_{XX}^{-1} exists because R_{XX} was assumed positive definite. Since W_* must satisfy the constraint (2.12)

$$C^TW_* = \mathcal{F} = C^TR_{XX}^{-1}R_{Xd} - C^TR_{XX}^{-1}C\lambda \quad (2.15)$$

and from (2.15) λ is found to be

$$\lambda = [C^TR_{XX}^{-1}C]^{-1}[C^TR_{XX}^{-1}R_{Xd} - \mathcal{F}] \ . \quad (2.16)$$

It is shown in Appendix C that the existence of $[C^TR_{XX}^{-1}C]^{-1}$ follows from the facts that R_{XX} is positive definite and

C has full rank. Substituting the last expression for the Lagrange multipliers into the expression for W_* (2.14) the result follows.

This completes the proof of Theorem 1.

C. Special Cases

A well-known special case of Theorem 1 is the unconstrained least-squares problem.

Corollary 1.1. (Least-Mean-Square Error -- Wiener) The optimum set of weights W_* solving the problem defined by Theorem 1 without constraints, i.e.,

$$\text{minimize} \quad E\{[d - W^T X]^2\} \tag{2.17}$$

with

$$E\{XX^T\} = R_{XX}$$

$$E\{Xd\} = R_{Xd}$$

is

$$W_{*_1} = R_{XX}^{-1} R_{Xd} \ . \tag{2.18}$$

And the best unconstrained estimate of d is $y = W_{*_1}^T X$.

Proof of Corollary 1.1.

Let the constraint matrix C vanish in Theorem 1. See especially Eq. (2.14) of the proof.

This completes the proof of Corollary 1.1.

A second well-known problem that can be formulated as a special case of Theorem 1 is the distortionless least-mean-squares estimation problem that was solved by Gauss.

Corollary 1.2. (Least-Mean-Squares Distortionless Estimate -- Gauss, Markov) Let the data vector X be of the form

$$X = CB + N , \tag{2.19}$$

where C is a known $(n \times m)$ matrix of rank m, and B is an unknown m-dimensional vector with $B^T = \underline{b_1 b_2 \ldots b_m}$. B may be a vector of random variables with unknown mean (so $E\{B\} = \overline{B}$), or it may be a vector of unknown parameters, in which case $E\{B\} = \overline{B} = B$. N is an unknown n-dimensional vector of random variables considered as noise. B and N are uncorrelated, with

$$\begin{aligned} E\{BB^T\} &= R_{BB} && (m \times m) \\ E\{N\} &= \theta && (n \times 1) \\ E\{NN^T\} &= R_{NN} && (n \times n) \\ E\{BN^T\} &= \underline{\underline{0}} && (m \times n) , \end{aligned}$$

and R_{NN} is positive definite.

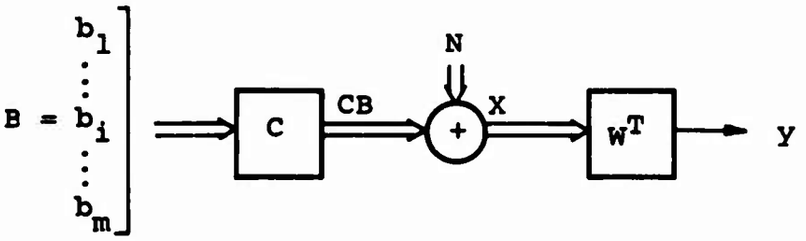
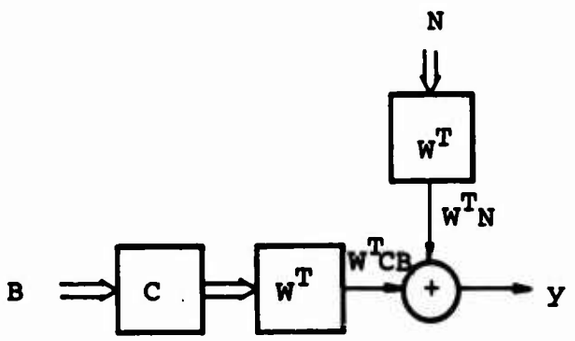
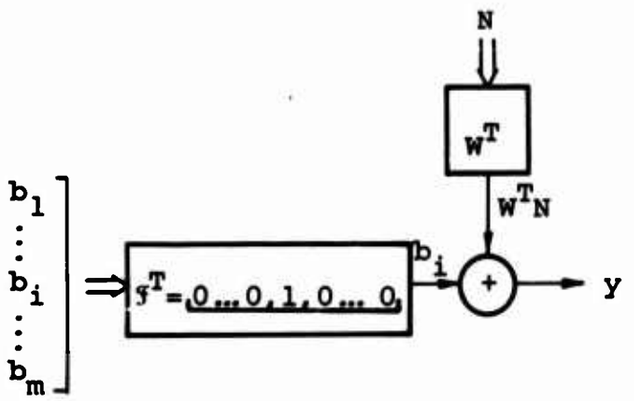


Fig. 2.1. The estimation problem of Corollary 1.2. Thick lines indicate vector-valued quantities. W is chosen so that y is an estimate of the i^{th} component of B, b_i.



(A)



(B)

Fig. 2.2. Manipulation of the flow charts from Fig. 2.1 yields (A). Constraining $W^TC = \xi^T$ yields (B), showing that the constraint puts a unity transfer function on b_i and that $y = b_i + W^TN$.

Thus

$$R_{XX} = CR_{BB}C^T + R_{NN} .$$

The problem is to make a linear least-squares estimate of b_i, say $y = W^T X$, that is unbiased (see Fig. 2.1). We wish the estimate of b_i to be corrupted only by the minimum amount of zero-mean noise. The optimum weight vector solving the problem

$$\begin{aligned} &\text{minimize} \quad E\{[b_i - W^T X]^2\} \\ &\text{subject to} \quad E\{W^T X - \bar{b}_i\} = 0 \end{aligned} \tag{2.20}$$

is

$$W_{*_2} = R_{XX}^{-1} C [C^T R_{XX}^{-1} C]^{-1} \mathfrak{s} \tag{2.21}$$

where

$$\mathfrak{s} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i^{th} \text{ position} \tag{2.22}$$

and the best unbiased estimate of b_i is $y = W_{*_2}^T X$.

Proof of Corollary 1.2.

The problem (2.20) is put into the form of the problem solved by Theorem 1. Observe that $b_i = \mathfrak{s}^T B$. Using (2.19) $X = CB + N$, and the fact that N is zero mean, we have

$$E\{W^T X - \bar{b}_i\} = E\{W^T CB + W^T N - \bar{b}_i\} = E\{W^T CB - \bar{b}_i\} .$$

Now if we require $C^T W = \mathfrak{s}$ then

$$E\{W^T X - \bar{b}_i\} = E\{\mathcal{F}^T B - \bar{b}_i\} = 0 \qquad (2.23)$$

and the constraint of (2.20) is satisfied. This is reasonable since the constraint applies a unit transfer function to b_i (see Fig. 2.2).

Further, with the constraint $C^T W = \mathcal{F}$ in force $y = W^T X = W^T C B + W^T N = b_i + W^T N$ and the cost function becomes

$$\begin{aligned} E\{[b_i - y]^2\} &= E\{b_i^2\} - 2E\{b_i y\} + E\{y^2\} \\ &= E\{b_i^2\} - 2E\{b_i (b_i + W^T N)\} + E\{y^2\} \\ &= E\{y^2\} - E\{b_i^2\} \ . \end{aligned} \qquad (2.24)$$

Because $E\{b_i^2\}$ is a constant, the weight vector that minimizes $E\{y^2\}$ also minimizes $E\{y^2\} - E\{b_i^2\}$, so the problem (2.20) reduces to

$$\begin{aligned} &\text{minimize} \quad E\{y^2\} = E\{[W^T X]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \ , \end{aligned} \qquad (2.25)$$

where C is defined in (2.19), and $\mathcal{F}$ in (2.22). This is a special case of the problem of Theorem 1 with $d = 0$. Since $d = 0$, $E\{Xd\} = R_{Xd} = \theta$ and so the first term of (2.8) vanishes and the optimal solution becomes the second term.

This completes the proof of Corollary 1.2.

In some cases the unbiased estimator may not be the most desirable. Suppose that (as in the array-processing problem discussed later) B is the sum of two vectors

$$B = \mathcal{A} + A \ , \qquad (2.26)$$

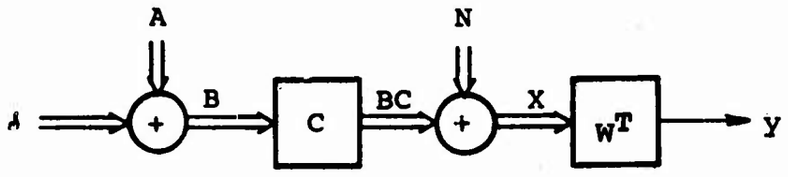


Fig. 2.3. B is a given structure for Corollary 1.3. B is the sum of signals δ plus noise A.

where $\mathcal{S}$ and $\mathcal{A}$ are m-dimensional vectors of random variables which may be statistically correlated. $\mathcal{S}$ is to be thought of as a vector of "signals", one of which we wish to estimate, say s_i. $\mathcal{A}$ is to be considered as a vector of additive noise. Note that $\mathcal{A}$ and $\mathcal{N}$ are both "noise" vectors of different dimension (see Fig. 2.3).

Since the "unbiased estimator" of Corollary 1.2 forms an estimate of b_i, which is equal to s_i plus a_i, it may not be satisfactory as an estimator of s_i alone.

Another approach is to recall from Corollary 1.2 that by suitable choice of the constraints a vector $\mathcal{F}$ can be applied directly to $\mathcal{B}$. Therefore a "filter" vector $\mathcal{F}$ (which may be different from (2.22)) may be designed to use the correlation among the components of $\mathcal{S}$ and the (hopefully different) correlations among the components of $\mathcal{A}$, so that when $\mathcal{F}$ is applied to $\mathcal{B}$ it may enhance s_i in the output and discriminate against $\mathcal{A}$. This is exactly analogous to the use of a filter in the frequency domain to pass signals and discriminate against noises.

In the following, it is assumed that $\mathcal{F}$ is a vector chosen by the user. The best choice of $\mathcal{F}$ is a topic with which we do not wish to get deeply involved. An example of a choice of $\mathcal{F}$ is given in Example 3, Section VI. If the weight vector $\mathcal{W}$ is constrained to satisfy $\mathcal{C}^T\mathcal{W} = \mathcal{F}$, then the output is

$$y = \mathcal{W}^T\mathcal{X} = \mathcal{W}^T\mathcal{C}\mathcal{B} + \mathcal{W}^T\mathcal{N} = \mathcal{F}^T\mathcal{B} + \mathcal{W}^T\mathcal{N} , \qquad (2.27)$$

and output power is

$$E\{y^2\} = \mathcal{F}^T R_{BB} \mathcal{F} + W^T R_{NN} W \quad . \qquad (2.28)$$

Because B and N are uncorrelated there is no cross term and so long as W satisfies the constraint any permissible variation in W affects only the power of the noise in the output. Thus the "degrees of freedom" of W not constrained by $C^T W = \mathcal{F}$ may be used to minimize the excess noise power in the estimate of s_i .

With the preceding motivation, the problem is set up as a special case of Theorem 1.

Corollary 1.3. (Least-Mean-Squares Filtered Estimate)

Let X be a known n-dimensional vector of observations of the form

$$X = CB + N \quad ,$$

where C is a known $(n \times m)$ matrix with $(n > m)$. B and N are unknown vectors of random variables with dimensions m and n respectively. Let B be of the form

$$B = \mathcal{A} + A \quad ,$$

where $\mathcal{A}$ and A are m-dimensional vectors of random variables. We wish to form an estimate of the i^{th} element of $\mathcal{A}$, s_i .

$$E\{N\} = \theta$$
$$E\{NN^T\} = R_{NN}$$
$$E\{BN^T\} = \underline{0}$$
$$E\{XX^T\} = R_{XX} .$$

Let $\mathcal{f}$ be a given m-dimensional filter vector. The least-mean-squares filtered estimator is the weight vector solving the problem

$$\begin{aligned} &\text{minimize} \quad E\{y^2\} = E\{[W^TX]^2\} \\ &\text{subject to} \quad C^TW = \mathcal{f} \end{aligned} \tag{2.29}$$

and is

$$W_{*_3} = R_{XX}^{-1}C[C^TR_{XX}^{-1}C]^{-1}\mathcal{f} \tag{2.30}$$

The best $\mathcal{f}$-filtered estimate of s_i is $y = W_{*_3}^T X$.

Proof of Corollary 1.3.

Proof follows directly from Theorem 1, with d=0 and hence $R_{Xd} = \theta$. The fact that $y = W_{*_3}^T X$ is an estimate for s_i follows from the above discussion.

This completes the Proof of Corollary 1.3.

Remark: If $\mathcal{f}$ is chosen as in (2.22) (one unit entry and the rest zeros) the solution is the same as the solution of Corollary 1.2.

III. THE ADAPTIVE ALGORITHM

A. The Unknown Statistics Problem

Suppose now that the correlation matrices R_{XX} and R_{Xd} required by Theorem 1 are not known a priori. Instead a sequence of observation vectors $\{X(0),X(1),\ldots,X(k),\ldots\}$ is presented, each vector drawn independently from a quasi-stationary ergodic distribution with autocorrelation R_{XX}. A sequence of random variables $\{d(0),d(1),\ldots,d(k),\ldots\}$ which are related to the X's by an unknown correlation vector R_{Xd} is also presented. We wish to minimize the constrained mean square error of the problem of Theorem 1.

An obvious solution is to make estimates of the unknown correlation matrices from observations, e.g.,

$$\hat{R}_{XX}(k) = \alpha\hat{R}_{XX}(k-1) + (1-\alpha)X(k-1)X^{T}(k-1) ,$$

and

$$\hat{R}_{Xd}(k) = \alpha\hat{R}_{Xd}(k-1) + (1-\alpha)X(k-1)d(k-1) ,$$

$$0 < \alpha < 1 ,$$

and insert these estimates into the expression for the optimal weight vector given by Theorem 1, Eq. (2.8). Inspection of (2.8) shows that because of the number of matrix multiplications and inversions involved, a great deal of computation is required at each iteration by this approach, ultimately limiting the rate at which estimates can be made and the dimensionality of a system of given cost. See Appendix F for an example of the performance of this approach.

The next section describes a computationally simple procedure (the Constrained-LMS algorithm) that converges to the weight vector W_* that solves the problem posed by Theorem 1 without prior knowledge of the correlation matrix R_{XX}. Further, if $d(k)$ is available for training, or is not required for the solution (as in Corollaries 1.2 and 1.3) then the algorithm does not require knowledge of R_{Xd}.

B. Derivation

The Constrained-LMS algorithm is based on a constrained gradient descent, satisfying $C^TW = \mathfrak{f}$ at all times while iterating to find a weight vector minimizing the cost function

$$J(W) = \frac{1}{2}E\{[d(k) - W^TX(k)]^2\} = \frac{1}{2}[Ed^2 - 2W^TR_{Xd} + W^TR_{XX}W] .$$

For motivation of the derivation, temporarily suppose that R_{XX} and R_{Xd} are known. As in the proof of Theorem 1, form the function $H(W)$ by adjoining the constraint to the cost function by a m-dimensional vector of Lagrange multipliers λ :

$$H(W) = \frac{1}{2}[Ed^2 - 2W^TR_{Xd} + W^TR_{XX}W] + \lambda^T[C^TW - \mathfrak{f}] . \qquad (3.1)$$

As in Theorem 1, we wish to find a weight vector W_* such that the gradient of H at W_* is θ and W_* satisfies $C^TW = \mathfrak{f}$. The gradient descent is initialized by choosing a weight vector $W(0)$ that satisfies the constraint.

The gradient of H with respect to W is

$$\nabla_W H = R_{XX}W - R_{Xd} + C\lambda \quad . \tag{3.2}$$

At each iteration the weight vector is moved in the direction of the negative gradient. (Note: a move in the direction of the positive gradient tends to increase a cost function.) The length of the step is proportional to the magnitude of the gradient and scaled by a gain factor μ . At the k^{th} iteration the next weight vector would be

$$\begin{aligned} W(k+1) &= W(k) - \mu\nabla_W H(k) \\ &= W(k) - \mu[R_{XX}W(k) - R_{Xd} + C\lambda(k)] \quad . \end{aligned} \tag{3.3}$$

The constrained gradient $[R_{XX}W(k) - R_{Xd} + C\lambda(k)]$ is the unconstrained gradient

$$\nabla_W J = R_{XX}W(k) - R_{Xd} \quad , \tag{3.4}$$

plus the term $C\lambda(k)$. As noted in Appendix E and later in Section IV, the vector $C\lambda(k)$ is orthogonal to the constraint. By proper choice of $\lambda(k)$ the component of the unconstrained gradient normal to the constraint (and hence deviating from it) can be exactly cancelled. Thus the Lagrange multipliers are chosen by requiring $W(k+1)$ to satisfy the constraint $\mathcal{G} = C^T W$:

$$\mathcal{G} = C^T W(k+1) = C^T W(k) - \mu C^T R_{XX} W(k) + \mu C^T R_{Xd} - \mu C^T C\lambda(k) \quad , \tag{3.5}$$

and solving for the Lagrange multipliers for the k^{th} iteration,

$$\lambda(k) = (C^TC)^{-1}C^TR_{XX}W(k) - \frac{1}{\mu}(C^TC)^{-1}C^TW(k) - (C^TC)^{-1}C^TR_{Xd} , \tag{3.6}$$

where it is shown in Appendix C that the existence of $(C^TC)^{-1}$ follows from the fact that C has full rank. Inserting the Lagrange multipliers of (3.6) into the iterative equation (3.3) we have

$$W(k+1) = W(k) - \mu[I-C(C^TC)^{-1}C^T][R_{XX}W(k)-R_{Xd}] + C(C^TC)^{-1}[\mathcal{G}-C^TW(k)] . \tag{3.7}$$

The algorithm may be rewritten, defining the n-dimensional vector

$$F = C(C^TC)^{-1}\mathcal{G} \tag{3.8}$$

$$W(k+1) = [I - C(C^TC)^{-1}C^T][W(k) - \mu R_{XX}W(k) + \mu R_{Xd}] + F. \tag{3.9}$$

Equation (3.9) is a deterministic gradient-descent algorithm that converges to the optimal weight vector W_* of Theorem 1 for a suitably small choice of the gain μ (proof given in Section VI). However, it requires knowledge of the correlation matrices R_{XX} and R_{Xd}, which in this study are assumed unavailable <u>a priori</u>. But recall $R_{XX} = E\{X(k)X^T(k)\}$ and $R_{Xd} = E\{X(k)d(k)\}$, so an easily-available and simple approximation for R_{XX} at the k^{th} iteration is the outer product of the observation vector with itself: $X(k)X^T(k)$; likewise if $d(k)$ is available

a simple approximation for R_{Xd} at the k^{th} iteration is $X(k)d(k)$.† This substitution gives the stochastic algorithm

$$W(k+1) = [I-C(C^TC)^{-1}C^T][W(k) - \mu X(k)X^T(k)W(k) + \mu X(k)d(k)] + F , \tag{3.10}$$

which can be simplified using $y(k) = X^T(k)W(k)$ and $e(k) = d(k) - y(k)$ to

$$W(k+1) = [I-C(C^TC)^{-1}C^T][W(k) + \mu e(k)X(k)] + F . \tag{3.11}$$

Equation (3.11) is the Constrained-LMS algorithm. It is a stochastic gradient-descent algorithm satisfying the constraint that $C^TW(k) = \mathcal{G}$ at all times (check: $C^TW(k+1) = \mathcal{G}$). At each iteration it requires only the observations $X(k)$ (and $d(k)$ if required). No <u>a priori</u> knowledge of R_{XX} or R_{Xd} is needed. The most comples operation is the multiplication of a constant matrix times a vector, which is a substantial savings over the matrix multiplications and inversions required (either explicitly or implicitly) by a direct implementation of the optimal equations.

The algorithm was derived heuristically. Its convergence to the optimum, rate of convergence, and steady-state

---

†As mentioned previously, better, but more complex, estimates for R_{XX} are available, such as $\frac{1}{k+1}\sum X(i)X^T(i)$. See Saradis, <u>et al.</u> [24] for use of this estimate in another algorithm; and Mantey and Griffiths [18] for a closely related estimate. For discussion and use of simpler estimates, see Moschner [20], Lender [15], and Nuttall [21]. The use of $X(k)X^T(k)$ here is a compromise between algorithm complexity and performance and may be changed if desired.

performance are shown in Section V. The next section develops the theory of constrained gradient descent from a geometrical viewpoint.

IV. A GEOMETRICAL VIEW OF THE ALGORITHM

A geometrical interpretation of the Constrained-LMS algorithm (3.11) is now given. Results will be found that permit an easier and more intuitive derivation of the properties of the algorithm than would otherwise be possible. Readers interested in applications may skip to Section VI.

We start from basic definitions.

Definition (Subspace) Let α and β be real scalar numbers. A nonempty subset S of a vector space is called a subspace if every vector of the form $\alpha V + \beta W$ is in S whenever V and W are both in S.

Since a subspace must contain at least one element W, it must also include the zero vector θ because $0 \cdot W = \theta$. Thus every subspace includes the origin.

Let Σ be the set of all n-dimensional weight vectors satisfying the homogeneous form of the constraint equation $C^T W = \theta$.

$$\Sigma \triangleq \{W : C^T W = \theta\} \quad . \tag{4.1}$$

Then we have

Geometrical Property 1. The set $\Sigma = \{W : C^T W = \theta\}$ defined by the homogeneous form of the constraint equation is a subspace.

Proof of Geometrical Property 1.

Let V and Z be vectors in Σ. They must satisfy the equations $C^T V = \theta$ and $C^T Z = \theta$. Therefore for any constants α and β, the vector $Y = \alpha V + \beta Z$ also satisfies $C^T Y = \theta$, so the set Σ is a subspace.

This completes the proof of Geometrical Property 1.

Definition (Linear Variety) A linear variety is a translation of a subspace.

A linear variety L may be expressed by the set equation $L = S + U$, where S is a subspace and U is any vector in the linear variety. The linear variety L is said to be parallel to the subspace S.

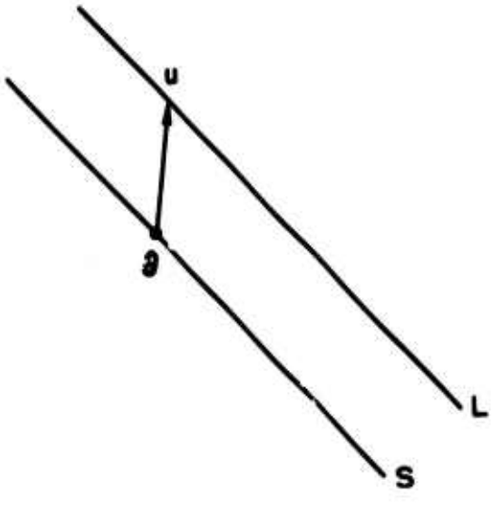


Fig. 4.1. A linear variety and its subspace.

Let Γ be the set of all weight vectors W satisfying the constraint $C^T W = \mathcal{F}$.

$$\Gamma \triangleq \{W : C^T W = \mathcal{F}\} . \qquad (4.2)$$

This definition leads to

Geometrical Property 2. The set $\Gamma = \{W : C^T W = \mathcal{F}\}$ defined by the constraint equation is a linear variety parallel to the subspace Σ .

Proof of Geometrical Property 2.

We must show that a vector W is in Γ if and only if it can be written as the sum of a vector in Σ and a translation vector.

(IF) Let the translation vector U be in Γ and Z be any vector in Σ . Then $C^T U = \mathcal{F}$ and $C^T Z = \theta$. Thus $W = Z + U$ satisfies $C^T W = C^T (Z + U) = \mathcal{F}$, so if U is in Γ the sum of any vector in Σ and U is in Γ .

(ONLY IF) Now suppose a vector W in Γ satisfied $C^T W = \mathcal{F}$ but could not be written as the sum of U and a vector in Σ . Then it follows that the vector $W - U$ could not be written as a vector in Σ . But $C^T (W - U) = \mathcal{F} - \mathcal{F} = \theta$ so $W - U$ is in Σ . Contradiction.

This completes the proof of Geometrical Property 2.

Geometrical Property 3. The shortest vector from the origin to the linear variety Γ is the vector $F = C(C^T C)^{-1} \mathcal{F}$, which is orthogonal to Γ .

Proof of Geometrical Property 3.

We want to find the vector W minimizing $\|W\|^2 = W^T W$ while satisfying $C^T W = \mathcal{F}$. Use the method of Lagrange

multipliers. Form the function $H(W)$ by adjoining the constraint to the cost criterion:

$$H(W) = \frac{1}{2}[W^T W] + \lambda^T (C^T W - \mathcal{F}) \ .$$

A necessary condition for optimality is

$$\nabla_W H = W + C\lambda = \theta \ ,$$

or

$$W = -C\lambda \ .$$

Requiring W to satisfy the constraint

$$C^T W = \mathcal{F}$$

we have

$$-C^T C\lambda = \mathcal{F} \ .$$

Solving for λ

$$\lambda = -(C^T C)^{-1}\mathcal{F} \ ,$$

and inserting this into the expression for W above

$$W = C(C^T C)^{-1}\mathcal{F} \ .$$

This is the vector F appearing in the algorithm (3.11) and defined in (3.8). As a check that F is in Γ note that $C^T F = C^T C(C^T C)^{-1}\mathcal{F} = \mathcal{F}$.

We wish to show F is orthogonal to Γ. Vectors parallel to the linear variety itself are the vectors of the parallel subspace Σ. Any vector Z in Σ

is orthogonal to $F = C(C^TC)^{-1}\mathfrak{z}$ since Z satisfies $C^TZ = \theta$ and so the inner product $F^TZ = \mathfrak{z}^T(C^TC)^{-1}C^TZ = 0$.

This completes the proof of Geometrical Property 3.

Note from the above proof that any vector of the form $C\gamma$, where γ is an m-vector, is orthogonal to the constraint variety Γ .

Geometrical properties 1- 3 are illustrated in Fig. 4.2.

The $(n \times n)$ matrix appearing between brackets in the algorithm (3.11) has an interesting geometrical interpretation. Call the matrix P .

$$P \triangleq I - C(C^TC)^{-1}C^T \quad . \tag{4.3}$$

The following definition appears in Luenberger [16]:

Definition. Let a vector W have a unique representation as the sum of two vectors, one from subspace Σ and the other from the subspace $\Sigma_\perp$ perpendicular to Σ . Thus let $W = W_{||} + W_\perp$, where $W_{||} \in \Sigma$, $W_\perp \in \Sigma_\perp$. The operator $\mathcal{P}$ defined by $\mathcal{P}W = W_{||}$ is called the projection operator onto Σ .

In other words, a projection operator acts as an identity operator on components in Σ and as a zero operator on components in $\Sigma_\perp$.

Geometrical Property 4. P is a projection operator onto Σ .

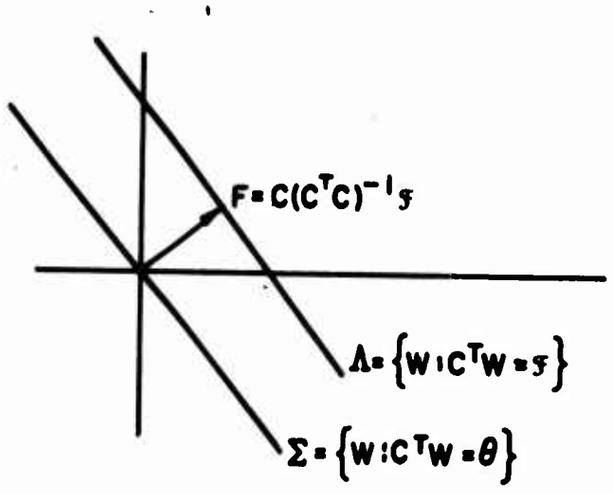


Fig. 4.2. The linear variety and subspace defined by the constraint.

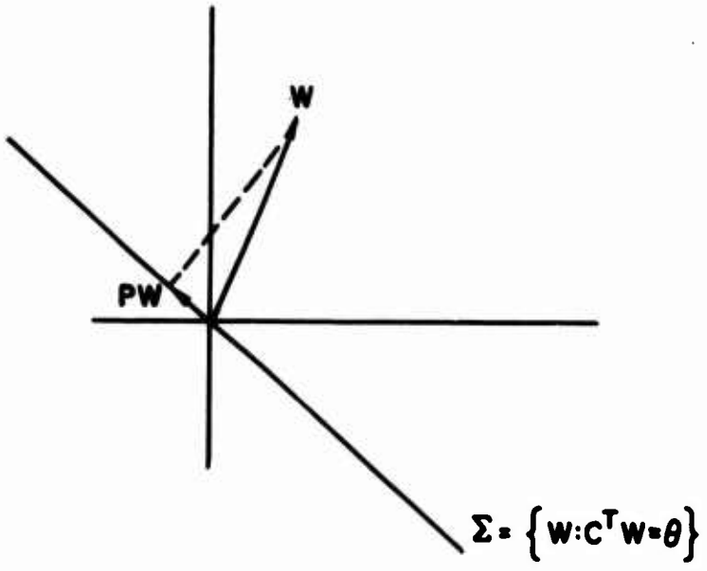


Fig. 4.3. P projects vectors onto Σ .

Proof of Geometrical Property 4.

Any weight vector W can be represented as

$$W = (W - PW) + PW .$$

The vector $PW = [I - C(C^TC)^{-1}C^T]W$ is in Σ since $C^T(PW) = C^T[I - C(C^TC)^{-1}C^T]W = [C^T - C^T]W = \theta$. The vector $(W - PW) = C(C^TC)^{-1}C^TW$ is in $\Sigma_\perp$. This is true because by definition of Σ for all vectors $Z \in \Sigma$, $C^TZ = \theta$; then every vector Z in Σ is orthogonal to $(W - PW)$ since $Z^TC(C^TC)^{-1}C^TW = \theta^T(C^TC)^{-1}C^TW = \theta$. Therefore we may make the identifications: $(W - PW) = W_\perp \in \Sigma_\perp$; $PW = W_\parallel \in \Sigma$; and $W = W_\parallel + W_\perp$, where $W_\parallel$ and $W_\perp$ satisfy the terms of the definition. By the second identification, P is the projection operator onto Σ .

This completes the proof of Geometrical Property 4.

The geometrical interpretation of P is shown in Fig. 4.3.

The algorithm (3.11) may be rewritten in terms of the projection operator:

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F . \qquad (4.4)$$

It should be mentioned that the vector $-e(k)X(k)$ is an estimate of the unconstrained gradient $\nabla_w J$. The unconstrained gradient, given in (3.4), is $R_{XX}W(k) - R_{Xd}$. Replacing R_{XX} by $X(k)X^T(k)$ and R_{Xd} by $X(k)d(k)$ results in $X(k)X^T(k)W(k) - X(k)d(k) = -e(k)X(k)$, where

$e(k) = d(k) - X^T(k)W(k)$. The algorithm is now considered as a whole.

The algorithm attempts to minimize the cost function $E\{[d(k) - W^T X(k)]^2\}$ by iterating to the optimal weight vector W_* along the constraint. Figure 4.4 shows the position of a hypothetical adaptive weight vector at iteration k and the position of the optimal weight vector.

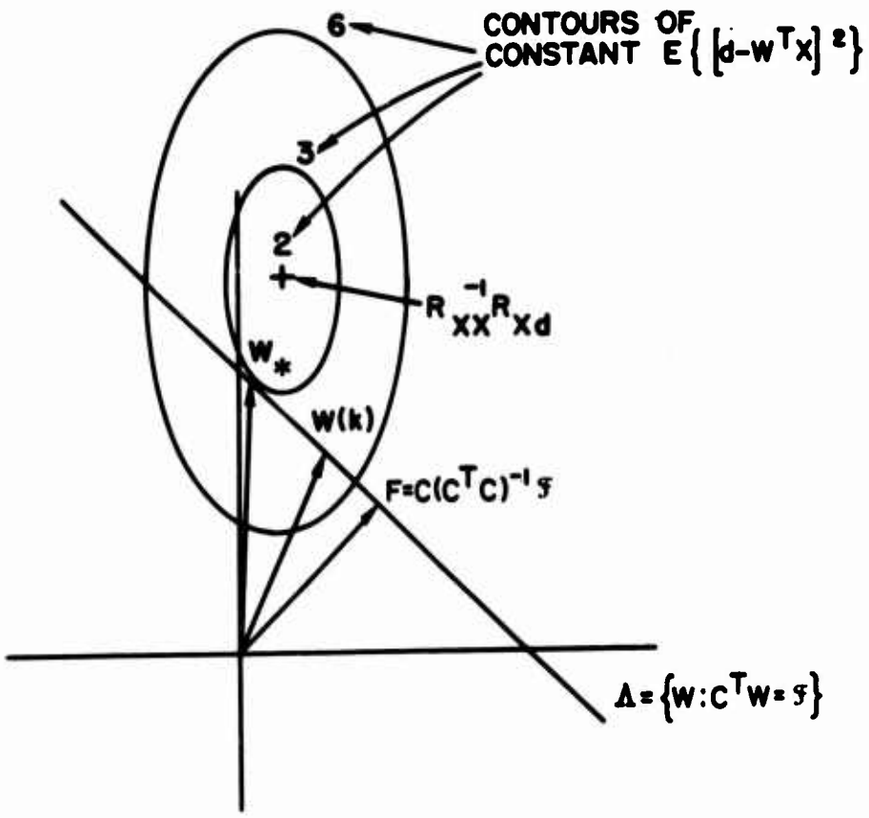


Fig. 4.4. Position of the adaptive weight vector $W(k)$ at the k^{th} iteration and the optimal constrained weight vector W_* .

$$W_* = [I - R_{XX}^{-1}C(C^T R_{XX}^{-1}C)^{-1}C^T R_{XX}^{-1}R_{Xd} + R_{XX}^{-1}C(C^T R_{XX}^{-1}C)^{-1}f \ .$$

The operation of the Constrained-LMS algorithm (4.4) is shown in Fig. 4.5. In this example, the unconstrained negative gradient estimate $e(k)X(k)$ is scaled by μ and added to the current weight vector. The resulting vector is projected onto the subspace Σ, producing a vector parallel to the constraint variety Λ. This vector is translated out to the constraint surface by adding it to F, forming the new weight vector $W(k+1)$ satisfying the constraint.

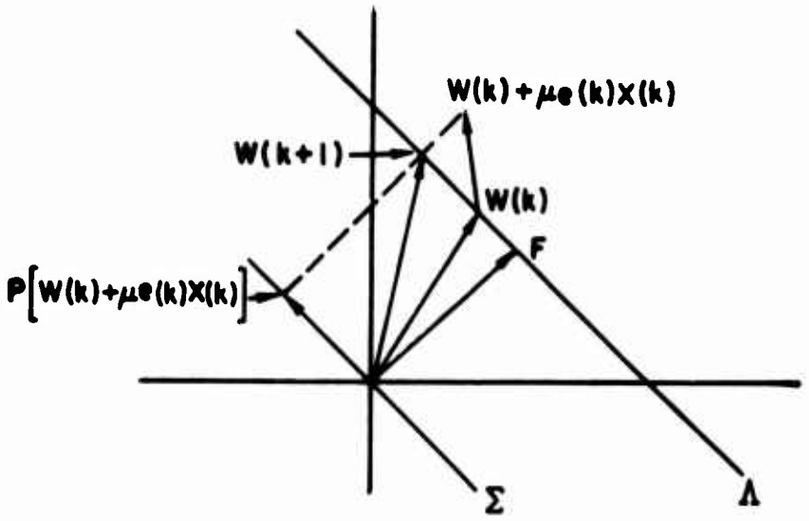


Fig. 4.5. Operation of the Constrained-LMS algorithm.

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F \ .$$

It is now shown that any difference vector between two vectors satisfying the constraint must lie in Σ (see Fig. 4.6). An identity that will be useful in the next section is given.

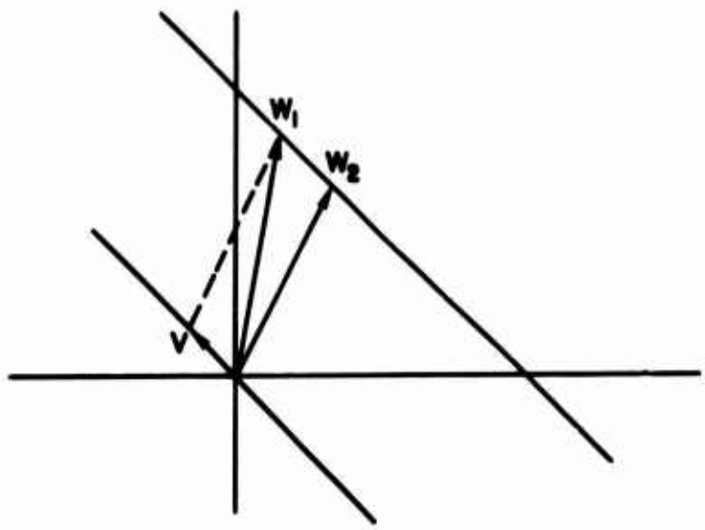


Fig. 4.6. The difference between two vectors satisfying the constraint is in the subspace Σ .

Geometrical Property 5. Let W_1 and W_2 be in Γ and let their difference be $V = W_1 - W_2$. Then V is in the subspace Σ and $PV = V$.

Proof of Geometrical Property 5.

Since W_1 and W_2 are in Γ, $C^T V = C^T W_1 - C^T W_2 = \mathcal{F} - \mathcal{F} = \theta$, so V is in Σ. By definition of a projection operator, if $V \in \Sigma$ then $PV = V$. Algebraically,

$PV = [I - C(C^T C)^{-1} C^T]V = V + \theta = V$.

This completes the proof of Geometrical Property 5.

Note also that P is symmetric and idempotent, i.e.,

$$P^T = P , \tag{4.5}$$

and

$$P^2 = P . \tag{4.6}$$

These are verified by carrying out the operations. The idempotence relation (4.6) for a matrix that is, in general, neither the zero nor the identity operator is interesting because it is impossible in the scalar case. It is a result of the fact (not proven here) that P has only zero and unity eigenvalues.

V. PERFORMANCE

In Part A of this section it is shown that the mean adaptive Constrained-LMS weight vector converges to the optimal constrained weight vector of Theorem 1. Rates of convergence along the eigenvectors of the matrix $PR_{XX}P$ are given. In part B it is shown that the difference in steady-state performance between the algorithm and the optimal estimator can be made arbitrarily small by decreasing the adaptive gain constant μ.

A. Convergence in Mean to the Optimum and Rate of Convergence

The algorithm (4.4) is repeated here in a more convenient form:

$$W(k+1) = P[W(k) - \mu X(k)X^T(k)W(k) + \mu X(k)d(k)] + F \quad . \quad (5.1)$$

Note that the weight vector $W(k)$ is a function of $W(0)$, $\{X(k-1),X(k-2),\ldots,X(0)\}$ and $\{d(k-1),d(k-2),\ldots,d(0)\}$. It was assumed at the beginning of Section III that the observation vectors X are independent,† so $X(k)$ is independent of $W(k)$. Taking the expected value of both sides of (5.1) we have an iterative equation in the mean value of the Constrained-LMS weight vector

$$EW(k+1) = P[EW(k) - \mu R_{XX}EW(k) + \mu R_{Xd}] + F \quad . \quad (5.2)$$

†This is believed to be an overly-restrictive assumption but greatly simplifies the analysis. For a special case of the algorithm (no constraints), Daniell [6] has shown ϵ-convergence assuming that the X's are only asymptotically independent.

Convergence of the mean is easily established using identities for expressing F and R_{Xd} in terms of the optimal weight vector:

$$F = [I - P]W_* \ , \tag{5.3}$$

$$R_{Xd} = PR_{XX}W_* \ , \tag{5.4}$$

both of which are verified directly using (2.4), (4.3), and (3.8). Let $V(k+1)$ be the difference between the mean adaptive weight vector at iteration $k+1$ and the optimal weight vector:

$$V(k+1) \triangleq EW(k+1) - W_* \ . \tag{5.5}$$

From (5.2)-(5.5) an equation for the difference process may be constructed:

$$\begin{aligned} V(k+1) &= P[EW(k) - \mu R_{XX}EW(k) + \mu PR_{XX}W_*] + [I - P]W_* - W_* \\ &= PV(k) - \mu PR_{XX}V(k) \ . \end{aligned} \tag{5.6}$$

Using $PV = (VP)^T = V$ from Geometrical Property 5 and (4.5) obtain

$$\begin{aligned} V(k+1) &= [I - \mu PR_{XX}P]V(k) \\ &= [I - \mu PR_{XX}P]^{k+1}V(0) \ . \end{aligned} \tag{5.7}$$

The matrix $PR_{XX}P$ is the correlation matrix of projected observations, i.e., $E\{(PX)(PX)^T\}$. The non-zero eigenvalues of this matrix are extremely important in determining both

the convergence rate of the algorithm and its steady-state performance relative to the optimum. The matrix being $(n \times n)$ and symmetric is diagonalizable into n orthogonal eigenvectors. It is shown in Appendix C that m of the eigenvectors of $PR_{XX}P$ lie entirely outside the subspace Σ and have zero eigenvalues; the other $(n-m)$ eigenvectors lie entirely within Σ and have strictly non-zero eigenvalues. All of the "action" is in the subspace Σ.

Call the $(n-m)$ non-zero eigenvalues of $PR_{XX}P$ $\sigma_i, i=1,2,\ldots,(n-m)$, and call the n (non-zero) eigenvalues of R_{XX} $\lambda_i, i=1,2,\ldots,n$. To get a feeling for the relationship between the σ's and the λ's, it is proven in Appendix C that the non-zero eigenvalues of $PR_{XX}P$ all fall between the largest and smallest eigenvalues of R_{XX}, that is, for $1 \leq i \leq (n-m)$

$$\lambda_{min} \leq \sigma_{min} \leq \sigma_i \leq \sigma_{max} \leq \lambda_{max}, \qquad (5.8)$$

where the subscripts min and max denote respectively the smallest and largest members of a set.

Since $V(0)$ is the difference between two vectors satisfying the constraint (5.5), from Geometrical Property 5 $V(0)$ lies entirely within the subspace Σ and may therefore be expressed as a linear combination of eigenvectors of $PR_{XX}P$ corresponding to non-zero eigenvalues. If $V(0)$ is equal to an eigenvector of $PR_{XX}P$, e_i with eigenvalue σ_i then

$$V(k+1) = [I - \mu PR_{XX}P]^{k+1} e_i$$

$$= [1 - \mu\sigma_i]^{k+1} e_i \ . \qquad (5.9)$$

Thus the convergence along any eigenvector of $PR_{XX}P$ is geometric with geometric ratio $[1 - \mu\sigma_i]$ and associated time constant

$$\tau_i = -1/\ln(1 - \mu\sigma_i) \cong 1/\mu\sigma_i \ , \qquad (5.10)$$

where the approximation is valid for $\mu\sigma_i \ll 1$. It is clear then that if μ is chosen so that

$$0 < \mu < 1/\sigma_{max} \ , \qquad (5.11)$$

then the euclidean norm of the difference vector is bounded between two ever-decreasing geometric progressions

$$[1 - \mu\sigma_{max}]^{k+1} \|V(0)\| \leq \|V(k+1)\| \leq [1 - \mu\sigma_{min}]^{k+1} \|V(0)\| \qquad (5.12)$$

and the expected value of the weight vector converges to the optimum with time constants given by (5.10) if the initial difference is finite.

We emphasize that convergence of the mean shown here is

$$\lim_{k \to \infty} \|EW(k) - W_*\| = 0 \ . \qquad (5.13)$$

B. Steady-state Performance Compared to Optimum

In this subsection the performance of the Constrained-LMS algorithm is compared with the optimum of Theorem 1 after transients have become negligible.

To allow the Constrained-LMS algorithm to operate in quasi-stationary (i.e., slowly time-varying) environments, the adaptive gain μ remains constant during the application of the algorithm. (In stochastic approximation schemes the gain is usually allowed to go to zero as time passes.) As a result of continually adapting, the weight vector has a non-zero variance about its optimal value. In a stationary noise field, the effect of variations about the optimum weight vector is to add a slight additional cost in excess of that achievable by the optimum. (See Brown [2] for results on time-varying noise fields.)

The excess cost normalized by the optimum cost level is a dimensionless quantity called "misadjustment" by Widrow [28] and is a measure of how closely the algorithm's performance achieves the optimal performance. Steady-state misadjustment is

$$M(\mu) = \lim_{k\to 0} \frac{\left[\begin{array}{c}\text{Cost of}\\ \text{Adaptive Filter}\\ \text{at time } k\end{array}\right] - \left[\begin{array}{c}\text{Cost of}\\ \text{Optimal Filter}\end{array}\right]}{\left[\begin{array}{c}\text{Cost of}\\ \text{Optimal Filter}\end{array}\right]} .$$

For the constrained least-mean-squares problem of Theorem 1 the steady-state misadjustment is

$$M(\mu) = \lim_{k\to 0} \frac{E\{[d(k) - W^T(k)X(k)]^2\} - E\{[d(k) - W_*^T X(k)]^2\}}{E\{[d(k) - W_*^T X(k)]^2\}} \qquad (5.14)$$

Under the assumptions that $d(k)$ and the components of $X(k)$ are jointly Gaussian-distributed and independent from observation to observation it is possible to calculate very tight bounds on $M(\mu)$ by a method due to Moschner [20]. For an adaptive gain constant satisfying

$$0 < \mu < \frac{1}{\sigma_{max} + (1/2)\mathrm{Tr}(PR_{XX}P)}, \qquad (5.15)$$

it is shown in Appendix B that steady-state misadjustment may be bounded by

$$\frac{\mu}{2} \frac{\mathrm{Tr}(PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{Tr}(PR_{XX}P) + 2\sigma_{min}]} \leq M(\mu) \leq \frac{\mu}{2} \frac{\mathrm{Tr}(PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{Tr}(PR_{XX}P) + 2\sigma_{max}]} \qquad (5.16)$$

$M(\mu)$ can be made arbitrarily close to zero by suitably small choice of gain constant μ ; this means that the steady-state performance of the Constrained-LMS algorithm can be made arbitrarily close to the optimum. From (5.10) it is seen that such cost performance is obtained at the expense of increased convergence time.

VI. APPLICATIONS

In this section adaptive solutions are given to the problems defined in Theorem 1 and its corollaries. At the same time the performance of each adaptive algorithm is given. The important application to array processing is the main example of this section.

The results of the preceding section are summarized in a companion theorem to Theorem 1:

Theorem 2. (Adaptive Constrained Least-Mean-Squares Optimization) Let $\{d(k)\}$ be a sequence of random variables and $\{X(k)\}$ be a sequence of n-dimensional data vectors of observed random variables. Each vector $X(k)$ is assumed to be produced independently by an unknown ergodic source with unknown correlation matrices

$$E\{X(k)X^T(k)\} = R_{XX} \qquad (n \times n)$$

$$E\{X(k)d(k)\} = R_{Xd} \qquad (n \times 1)$$

and R_{XX} positive definite. The algorithm

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F \ , \qquad (6.1)$$

where

$$P = [I - C(C^TC)^{-1}C^T] \ ,$$

$$F = C(C^TC)^{-1}\mathcal{F} \ ,$$

$$C^TW(0) = \mathcal{F} \ ,$$

and

$$e(k) = d(k) - W^T(k)X(k) ,$$

converges in the mean to the optimum weight vector W_* solving the constrained LMS problem defined in Theorem 1:

$$\begin{aligned} &\text{minimize} \quad E\{[(d(k) - W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} , \end{aligned} \tag{6.2}$$

if

$$0 < \mu < \frac{1}{\sigma_{max} + \frac{1}{2} \operatorname{Tr}(PR_{XX}P)} . \tag{6.3}$$

Further,

i) the convergence time constant of the difference between $EW(k)$ and W_* along the i^{th} eigenvector of $PR_{XX}P$ is

$$\tau_i = \frac{-1}{\ln(1 - \mu\sigma_i)} \cong 1/\mu\sigma_i \tag{6.4}$$

where σ_i is the eigenvalue corresponding to the i^{th} eigenvector of $PR_{XX}P$.

ii) Under the additional assumption that variables $d(k)$ and $X(k)$ are jointly Gaussian distributed, the steady-state misadjustment of the adaptive solution can be bounded by (5.16).

Proof of Theorem 2. (See Section V)

This completes the proof of Theorem 2.

Example 1. (Consistent Modeling)

A single-input, single-output system is to be modeled with a tapped-delay-line filter. It is known that the system's steady-state response to a unit step-function input is a particular number α, and this feature is to be incorporated into the model.

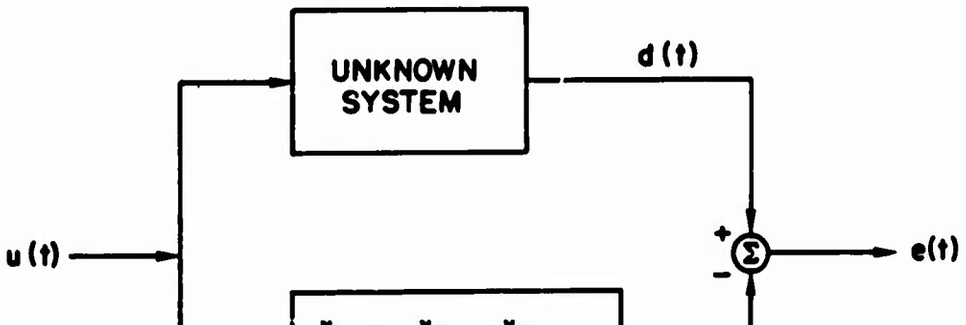


Fig. 6.1. Tapped-delay-line filter modeling a system.

Let the input to the system and model be a random variable $u(t)$. The model is a tapped-delay-line filter with n tap points and a delay of Δ seconds between each tap. Inputs pass down the tapped-delay-line (TDL) filter. Let the states at each tap point be denoted $x_i(t), i=1,2,\ldots,n$.

Thus the first state is equal to the input, $x_1(t) = u(t)$, the second state is equal to the input delayed by Δ seconds, $x_2(t) = u(t-\Delta)$, and so forth. The output $y(t)$ of the TDL filter is a weighted sum of the states. Let the weight on the i^{th} state be w_i and form the n-dimensional vectors of weights $W^T = (w_1, w_2, \ldots, w_n)$ and states $X^T(t) = (x_1(t), x_2(t), \ldots, x_n(t))$. The output of the TDL filter at time t is then $y(t) = W^T X(t)$. The desired output of the model is the output of the unknown system $d(t)$. The error is the difference between the desired output and the actual output: $e(t) = d(t) - y(t)$.

The constraint is now included. If the systems are given a unit step input (i.e., $u(t) = 1$), then after $n\Delta$ seconds the TDL filter will be in steady state, with $X^T(t) = \underline{1}^T = (1, 1, \ldots, 1)$. Thus the constraint that the steady-state response of the TDL filter be a α is equivalent to requiring

$$\underline{1}^T W = \alpha \ . \tag{6.5}$$

The consistent modeling problem is therefore

$$\begin{aligned} &\text{minimize} \quad E\{e^2(t)\} \\ &\text{subject to} \quad \underline{1}^T W = \alpha \ . \end{aligned} \tag{6.6}$$

To apply the Constrained LMS algorithm, the state vector and error are sampled at intervals of T seconds. At the k^{th} sampling instant the state vector is $X(kT)$ and the error is $e(kT)$. The time between samples T is made large enough

so that $X(kT)$ is essentially independent of $X(jT)$ for $j \neq k$. (As noted in Section V this is not believed to be absolutely necessary). The algorithm is therefore

$$W(k+1) = P[W(k) + \mu e(kT)X(kT)] + \alpha \tag{6.7}$$

where

$$P = I - \underline{1}(\underline{1}^T\underline{1})^{-1}\underline{1}^T = I - \frac{1}{n}\underline{1}\underline{1}^T ,$$

and the weight vector is assumed to be constant for t in $kT \leq t < (k+1)T$.

A practical matter arises here. It may be difficult to calculate the permissible upper bound on μ given by (6.3), especially if the autocorrelation matrix R_{XX} is not known. An easily measured quantity guaranteed to be no higher than the permissible upper bound is

$$\mu_0 = \frac{1}{\frac{3}{2}\,\mathrm{Tr}(R_{XX})} , \tag{6.8}$$

That is, if μ is chosen to satisfy

$$0 < \mu < \mu_0 , \tag{6.9}$$

then it is guaranteed to satisfy (6.3). Observe that μ_0 can be calculated directly and easily from observations since $\mathrm{Tr}(R_{XX}) = E[X^T(k)X(k)]$, the sum of the powers of the states.

A special case of the algorithm given in Theorem 2 is the celebrated LMS algorithm. The following is a companion to Corollary 1.1.

Corollary 2.1. (Adaptive Least-Mean-Squares Optimization -- Widrow and Hoff) Let the sequences $\{d(k)\}$, $\{X(k)\}$ and their (unknown) correlation matrices be defined as in Theorem 2. The algorithm

$$W(k+1) = W(k) + \mu e(k)X(k) \tag{6.10}$$

where

$$e(k) = d(k) - W^T(k)X(k) \ .$$

converges in the mean to the optimum weight vector W_{*1} solving the unconstrained least-mean-squares optimization problem defined in Corollary 1.1:

$$\text{minimize} \quad E\{e^2(k)\} \ , \tag{6.11}$$

if

$$0 < \mu < \frac{1}{\lambda_{max} + \frac{1}{2} \operatorname{Tr}(R_{XX})} \ . \tag{6.12}$$

Further,

i) the convergence time constant of the difference between $EW(k)$ and W_{*1} along the i^{th} eigenvector of R_{XX} is

$$\tau_i = \frac{-1}{\ln(1 - \mu\lambda_i)} \cong 1/\mu\lambda_i \ , \tag{6.13}$$

where λ_i is the eigenvalue corresponding to the i^{th} eigenvector of R_{XX}.

ii) Under the additional assumption that variables $d(k)$ and $X(k)$ are jointly Gaussian distributed, the steady state misadjustment of the adaptive solution can be bounded by

$$\frac{\mu}{2}\frac{\mathrm{Tr}(R_{XX})}{1-\frac{\mu}{2}[\mathrm{Tr}(R_{XX})+2\lambda_{min}]} \leq M(\mu) \leq \frac{\mu}{2}\frac{\mathrm{Tr}(R_{XX})}{1-\frac{\mu}{2}[\mathrm{Tr}(R_{XX})+2\lambda_{max}]} \tag{6.14}$$

Proof of Corollary 2.1.

The projection operator P of Theorem 2 goes to the identity when all constraints are removed. The vector F vanishes.

This completes the proof of Corollary 2.1.

Uses of linear least-squares algorithms are abundant and well-known so no examples will be given.

The next corollary is a companion to Corollary 1.2.

Corollary 2.2. (Adaptive Least-Mean-Squares Distortionless Estimate) Let the sequences $\{d(k)\}$, $\{X(k)\}$, and their (unknown) correlation matrices be defined as in Theorem 2. Further, let each $X(k)$ be of the form

$$X(k) = CB(k) + N(k) \ , \qquad (6.15)$$

where C is a known $(n \times m)$ matrix with $n > m$ and $\{B(k)\}$ is a sequence of unknown m-dimensional vectors. Each $B(k)$ may be a vector of random variables with unknown mean (so $E\{B\} = \overline{B}$), or it may be a vector of unknown parameters, in which case $E\{B\} = \overline{B} = B$. $\{N(k)\}$ is a sequence of unknown n-dimensional zero-mean random vectors considered as noise. $B(k)$ and $N(k)$ are assumed uncorrelated. Let

$$B(k) = \begin{bmatrix} b_1(k) \\ b_2(k) \\ \vdots \\ b_m(k) \end{bmatrix} \quad \text{and} \quad \mathcal{F} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i^{th} \text{ component} \ . \qquad (6.16)$$

The algorithm

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F \qquad (6.17)$$

where

$$P = [I - C(C^TC)^{-1}C^T]$$

$$F = C(C^TC)^{-1}\mathcal{F}$$

$$C^TW(0) = \mathcal{F}$$

$$y(k) = W^T(k)X(k) \ ,$$

converges in the mean to the optimum weight vector W_{*2} solving the least-mean-squares distortionless estimation problem defined in Corollary 1.2:

$$\begin{aligned} &\text{minimize} \quad E\{[b_i - W^T X(k)]^2\} \\ &\text{subject to} \quad E\{(\bar{b}_i - W^T X(k)\} = 0 \ , \end{aligned} \tag{6.18}$$

as long as μ satisfies condition (6.3) of Theorem 2.

The convergence rates and misadjustment are the same as those of Eqs. (6.4) and (5.16) of Theorem 2.

Proof of Corollary 2.2.

It was shown in Corollary 1.2 that the problem (6.18) may be reformulated as

$$\begin{aligned} &\text{minimize} \quad E\{[W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \ . \end{aligned} \tag{6.19}$$

This is just the problem of Theorem 2 with $d(k) = 0$. Accordingly in the Constrained-LMS algorithm (6.1), $d(k)$ is set to zero and the corollary follows. The requirements on μ and performance are unchanged.

This completes the proof of Corollary 2.2.

Example 2. (State Estimation under Uncertainty)

A system is described by the equations

$$\begin{aligned} B(k+1) &= \Phi B(k) + \Gamma U(k) \\ X(k) &= C\,B(k) + N(k) \ , \end{aligned} \tag{6.20}$$

where C is a known $(n \times m)$ matrix with $n > m$. $B(k)$ is the state vector and $U(k)$ is the input vector. $N(k)$ is zero-mean measurement noise. The matrices Φ and Γ are

not known; the statistics of $U(k)$ and $N(k)$ are not known. We wish to estimate a component $b_i(k)$ of the state vector. The algorithm (6.17) converges to the best constant linear least squares unbiased estimator of a component of $B(k)$. If an estimate of the entire vector is desired, m algorithms like (6.1) may be used, with the unit entry in a different place in each $\mathcal{Y}$ vector.

■

Note that the amount of knowledge required for an estimate of $B(k)$ using the constrained LMS algorithm is a small fraction of that required by the Kalman filter. (The Kalman filter is the optimum unconstrained time-varying linear least-squares estimator for the state vector of the dynamic system (6.20), and requires that all system matrices and correlation matrices be known.)

The next corollary is a companion to Corollary 1.3.

Corollary 2.3. (Adaptive Least-Mean-Squares Filtered Estimate) Let the sequence $\{d(k)\}$, $\{X(k)\}$, and their (unknown) correlation matrices be defined as in Theorem 2. Let

$$X(k) = C\,B(k) + N(k), \tag{6.21}$$

$B(k)$ and $N(k)$ be vectors of random variables as in Corollary 1.3. Further, let $B(k)$ be written

$$B(k) = \mathcal{A}(k) + A(k), \tag{6.22}$$

where $\{\mathcal{A}(k)\}$ and $\{A(k)\}$ are sequences of m-dimensional vectors of random variables. We wish to estimate the

i^{th} component of $\mathcal{A}(k)$, $s_i(k)$. A filter vector $\mathcal{F}$ is given, which may be based on as much or as little information about the statistics of $\mathcal{A}(k)$ and $A(k)$ as is known.

The algorithm

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F , \tag{6.23}$$

where

$$P = [I - C(C^TC)^{-1}C^T]$$

$$F = C(C^TC)^{-1}\mathcal{F}$$

$$C^TW(0) = \mathcal{F}$$

$$y(k) = W^T(k)X(k) ,$$

converges in the mean to the optimum weight vector W_{*3} solving the least-mean-squares filtering problem given in Corollary 1.3:

$$\begin{aligned} &\text{minimize} \quad E\{[s_i(k) - W^TX(k)]^2\} \\ &\text{subject to} \quad C^TW = \mathcal{F} , \end{aligned} \tag{6.24}$$

as long as μ satisfies condition (6.3) of Theorem 2.

The convergence rates and misadjustment are the same as those of Eqs. (6.4) and (5.16) of Theorem 2.

Proof of Corollary 2.3.

In Corollary 1.3, it is shown that the problem (6.24) may be reformulated as

$$\text{minimize} \quad E\{[W^T X(k)]^2\}$$
$$\text{subject to} \quad C^T W = \mathcal{F} \quad . \qquad (6.25)$$

As before, this is just the problem of Theorem 2 with $d(k) = 0$. The results follow from Theorem 2.

This completes the proof of Corollary 2.3.

The next example is the major example of the paper, and is one of the main reasons for an interest in adaptive constrained least squares optimization: It is shown in this example that adaptive constrained LMS optimization makes possible the near-optimum processing of data from an array of antennas or other sensors with very little a priori information about the signals and noises involved. In contrast, known adaptive processors converging to the optimal unconstrained least-mean-squares filter [12] require knowledge of either the signal or noise statistics.

Example 3. (The Array Processor)

In most applications involving arrays of sensors -- notably sonar, seismology, radio communication, and radio astronomy using antenna arrays -- it is desirable to reduce antenna sensitivity to unwanted signals and noises while processing the signals of interest in real time. For example, arrays of sonar hydrophones provide information about the undersea environment; it may be desirable to listen to signals coming from a particular direction and simultaneously avoid hearing the noise of the sonar ship's own

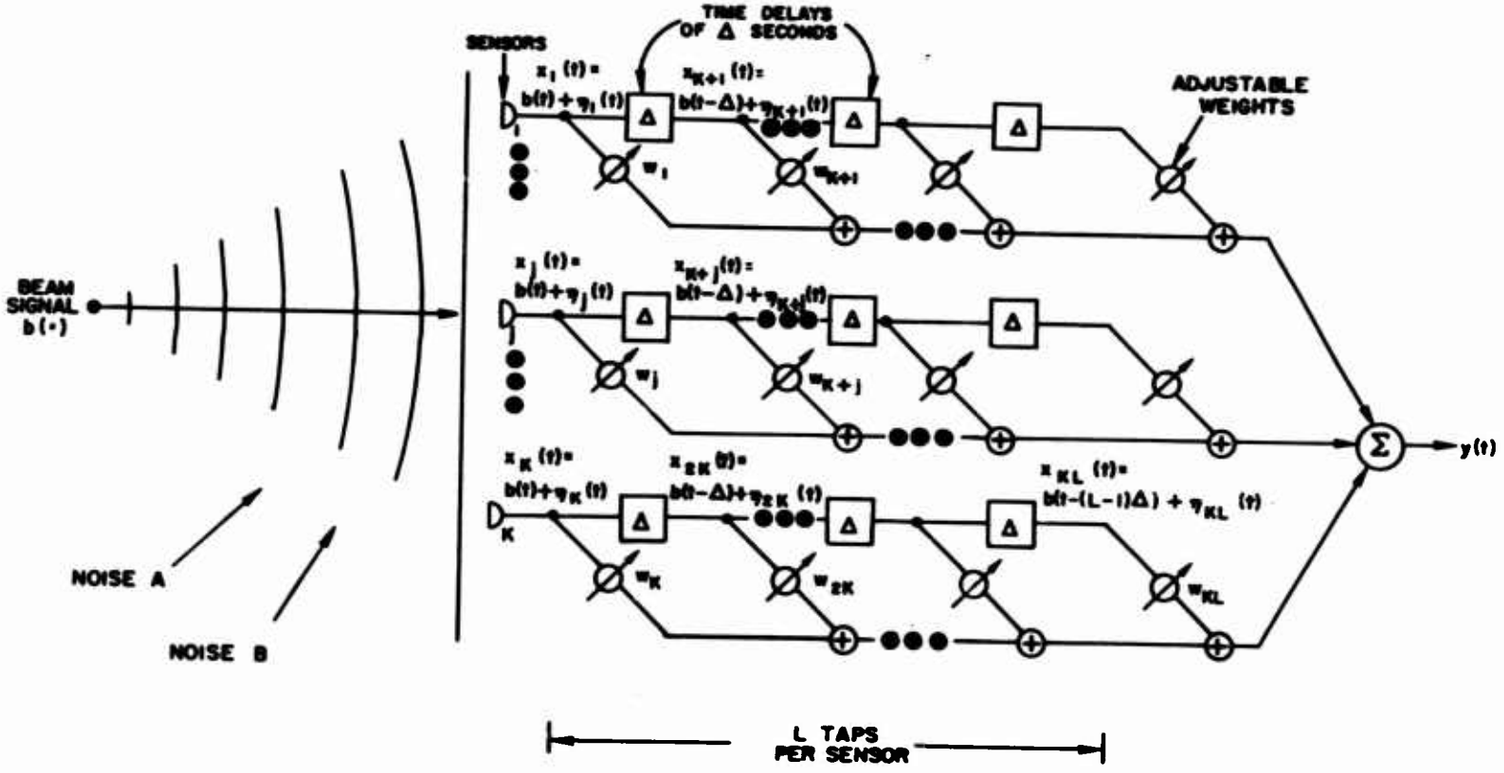


Fig. 6.2. Signals and noises incident on the array. Because the array is steered toward the look direction, all beam signal components on any given column of filter taps are identical.

machinery and screws [22]. In geology, sub-arrays of seismometers are being used in the large-aperture seismic array in Montana [4] to listen to seismic events; such arrays must discriminate against noises emanating from surrounding cities. In radio communications using antenna arrays, it is desirable to receive signals from one direction while ignoring the signals from amateur radio operators and other electromagnetic noises [5]. Radio astronomers using antenna arrays want to look in one part of the sky while discriminating against other radiation sources impinging on antenna sidelobes [25, p. 33]. In all these applications, it is desirable to have a processor that can discriminate against unwanted noises in real time and that requires a minimum of *a priori* information.

An array processor is a filter both in frequency and in space. A typical processor configuration is shown in Fig. 6.2. The array has K sensors with a tapped delay line following each sensor. Each line has L tap points and delays of Δ seconds between taps. Signals and noises impinging on the array are converted to voltages which pass down the tapped delay lines and the weighted sum of these voltages is the output of the filter. By proper choice of weights, the array processor can discriminate against unwanted noises distributed both in frequency and space.

The filter separates desirable signals from other noises. In this example, to be "desirable" a signal must come from

a particular chosen direction in space, called the "look direction". All signals coming from other directions, plus any measurement or amplifier noise, are termed "undesirable noise". But not all signals coming from the look direction are desirable; some noise comes from the look direction and is called "look direction noise".

The signal is modeled as a zero-mean random process emanating from the look direction in the far field of the array. It is assumed that the propagating medium is linear, non-dispersive, and that propagation times along the signal phase-front are well enough known that the array can be steered, electrically or mechanically, in the direction of the signal. Sources in the look direction, i.e., desired signal and look direction noise, are assumed to be statistically uncorrelated with noises emanating from other directions. (This rules out multipath.) Finally, all the sensors are assumed to have identical characteristics (but are not necessarily omnidirectional).

It should be mentioned that sonar and seismic signals are generally low frequency (audio or lower) and may be processed in real time using the adaptive algorithm implemented by present-day hardware [13]. In radio-frequency applications, however, the signals must first be demodulated.

As in Example 1, let the observations at each tap points at time t be denoted $x_i(t), i=1,2,\ldots,n$. In this case $n = KL$, the number of sensors times the number of taps per

sensor. Let the weight on the i^{th} observation be w_i and form the n-dimensional vectors of weights $W^T = (w_1, w_2, \ldots, w_n)$ and observations $X^T(t) = (x_1(t), x_2(t), \ldots, x_n(t))$. The output of the array processor at time t is then $y(t) = W^T X(t)$. Let the signals arriving "in the beam" (i.e., from the look direction) at time t be denoted $b(t)$. Because the array is steered toward the look direction, signals arriving "in the beam" enter each of the K tapped-delay lines simultaneously and parade in parallel down the lines (see Fig. 6.2). All K taps on the first column of taps have the same beam component, $b(t)$, and a different undesirable noise component $\eta_i(t), i=1, \ldots, K$ from noises entering from other directions and amplifier noise; every tap on the second column of taps has the same beam component $b(t-\Delta)$ and a different noise component $\eta_i(t), i=K+1, \ldots, 2K.$, and so forth. Forming the L-dimensional vector of beam signals on each column at time t , $B^T(t) = [b(t), b(t-\Delta), \ldots, b(t-(L-1)\Delta)]$ and the n-dimensional vector of undesirable noises on each tap at time t , $N^T(t) = [\eta_1(t), \eta_2(t), \ldots, \eta_n(t)]$ it is seen from Fig. 6.2 that the vector of observations may be written

$$X(t) = C\,B(t) + N(t) \quad , \tag{6.26}$$

where

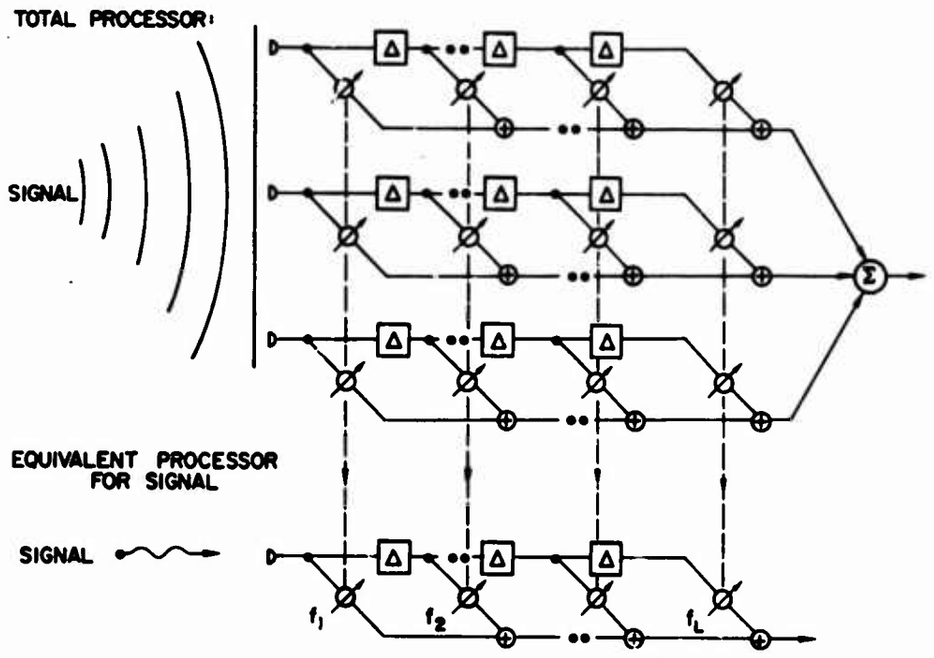


Fig. 6.3 Equivalent processor for signals coming from the look direction.

$$C = \begin{matrix} \overset{\longleftarrow L \longrightarrow}{} \\ \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} K \end{matrix} \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \\ 1 & 0 & 0 & & \cdot \\ 0 & 1 & 0 & & \cdot \\ & \vdots & & & \cdot \\ 0 & 1 & 0 & & \\ & & & \ddots & \\ & & & 0 & 1 \\ & & & \vdots & \vdots \\ 0 & \cdots & & 0 & 1 \end{bmatrix} \left.\begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix}\right\} KL \qquad (6.27)$$

Due to the array steering it is particularly easy to specify the frequency response of the processor in the look direction, since all K taps in every vertical column of taps have identical beam components: The processor output is formed by a weighted sum of the observations; therefore, as far as the beam signal is concerned, the total weighting it receives at each vertical column is the sum of the weights in that column (see Fig. 6.3). For the beam signal, the multichannel processor could be replaced by a single-channel filter. Each weight on the single-channel filter would be equal to the sum of weights on the corresponding column of the multichannel filter.

Let the i^{th} weight on the beam-signal-equivalent filter in Fig. 6.3 be f_i. Let the L-dimensional vector of filter weights be

$$\mathcal{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_L \end{bmatrix} . \qquad (6.28)$$

Each weight f_i is the sum of the weights in the i^{th} column of the multichannel processor. Referring to the definition of C above, it is seen that this statement is equivalent to

$$\mathcal{F} = C^T W . \qquad (6.29)$$

$\mathcal{F}$ determines the transfer function of the processor in the look direction. If, for example,

$$\mathcal{F}^T = \underline{(0,0,\ldots,0,1,0,\ldots,0)} , \qquad (6.30)$$

then all frequencies of signals arriving from the look direction in plane waves would be passed equally without attenuation (flat frequency response). Changing any of the zero components would result in a different impulse response and corresponding frequency shaping.

Recall that in the beam signal $b(t)$ there is a component of "desired signal" $s(t)$ and an additive "look direction noise" $a(t)$, i.e., $b(t) = s(t) + a(t)$. It is assumed that from any <u>a priori</u> information he might have about the frequency content of $s(t)$ or $a(t)$, the processor user specifies the look direction frequency response he wants in the form of the vector $\mathcal{F}$. If he has no prior

information about the desired signal then a reasonable choice for $\mathcal{F}$ is the all-pass filter (6.30). Notice that specifying the look direction frequency response constrains only L "degrees of freedom" of the n weights in W. The remaining "degrees of freedom" are used by the processor to reduce the power of undesirable noises $N(t)$ in the output. Since the response to the beam signals is constrained and the undesirable noises are assumed uncorrelated with the beam signal, minimizing total processor output power is exactly the same as minimizing noise output power.

The problem is

$$\begin{aligned} &\text{minimize} \quad E\{[W^T X(t)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \end{aligned} \tag{6.31}$$

and the algorithm is (6.22): $W(k+1) = P[W(k) - \mu y(kT)X(kT] + F$ where T is the time between adaptations, made sufficiently large so that successive vectors X are essentially independent.

In this case P is simple and sparse due to the simple form of the constraint matrix (6.26). The matrix multiplication by P is more simply regarded as a series of additions and scalar multiplications:

$$P = \begin{bmatrix} 1 & -\frac{1}{K} & \cdots & -\frac{1}{K} & 0 & & & 0 & & & & 0 \\ -\frac{1}{K} & 1 & & & & & & & & & & \\ \vdots & & \ddots & & & & & & & & & \\ -\frac{1}{K} & -\frac{1}{K} & \cdots & 1 & 0 & 0 & & 0 & & & & \\ 0 & 0 & & 0 & 1 & -\frac{1}{K} & \cdots & -\frac{1}{K} & 0 & \cdots & & \\ & & & 0 & -\frac{1}{K} & 1 & & & 0 & & & \\ & & & \vdots & & & \ddots & & & & & \\ & & & 0 & -\frac{1}{K} & -\frac{1}{K} & \cdots & 1 & 0 & & & \\ & & & & & & & & 1 & -\frac{1}{K} & \cdots & -\frac{1}{K} \\ & & & & & & & & -\frac{1}{K} & 1 & & \\ & & & & & & & & \vdots & & \ddots & \\ 0 & & & \cdots & & & & 0 & -\frac{1}{K} & -\frac{1}{K} & \cdots & 1 \end{bmatrix} \tag{6.32}$$

A computer simulation of the processor was made using a low-precision language (BASIC) on a small computer (the HP 2116). The processor had four sensors on a line spaced at Δ second intervals and had four taps per sensor (thus n=16). The environment had three point noise sources, and white noise added to each sensor. Power of the beam signal was quite small in comparison to the power of interfering noises (see Table 6.1). The tap spacing defined a frequency of

SOURCE	POWER	DIRECTION (0° IS NORMAL TO ARRAY)	CENTER FREQUENCY (1.0 is $1/\Delta$)	BANDWIDTH
Beam Signal	0.1	0°	0.3	0.1
Noise A	1.0	45°	0.2	0.05
Noise B	1.0	60°	0.4	0.07
White Noise (per tap)	0.1	-	-	-

Table 6.1. Signals and noises in the simulation

1.0 (i.e., f = 1.0 is a frequency of $1/\Delta$ Hz.). In the look direction, foldover frequency for the processor response was $1/2\Delta$, or 0.5. All signals were generated by a pseudo-random, pseudo-Gaussian generator and passed through a filter to give them the proper spatial and temporal correlations. All temporal correlations were arranged to be identically zero for time differences greater than 25Δ. The time between adaptations was assumed greater than 58Δ, so successive samples of $X(kT)$ were generated independently.

The look direction filter was specified by the vector $\underline{\phi}^T = \underline{1,-2,1.5,2,}$ which resulted in a frequency characteristic shown in Fig. 6.4. The signal and noise spectra are shown in Fig. 6.5 and their spatial position in Fig. 6.2.

In this problem, the eigenvalues of R_{XX} ranged from 0.111 to 8.355. The upper permissible bound on the gain constant μ calculated by (6.3) was .074; a value of $\mu = .01$ was selected, which, by (5.16) would lead to a misadjustment of between 15.2 and 17.0%.

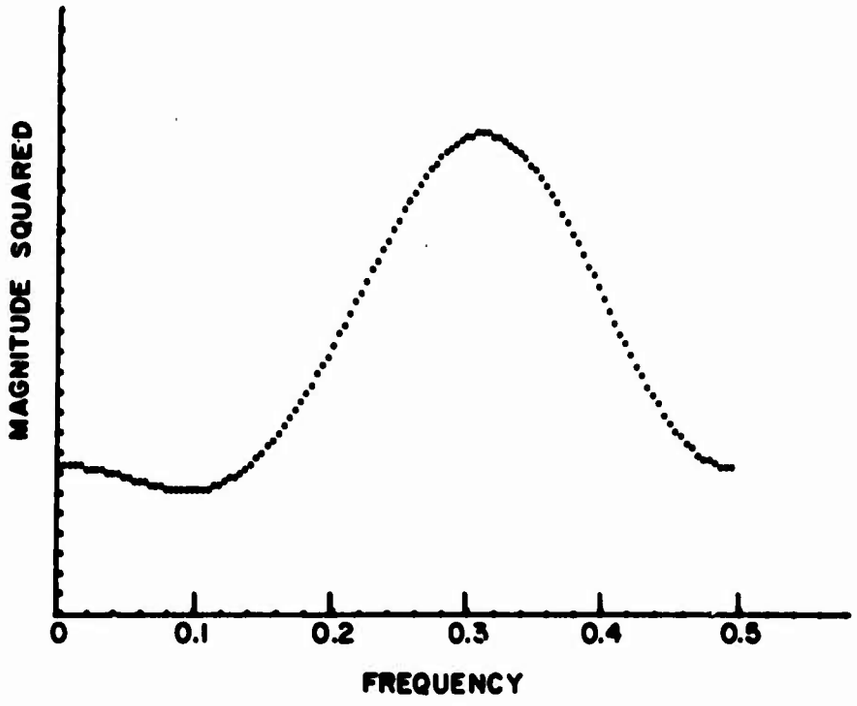


Fig. 6.4. Frequency response of the processor in the look direction.

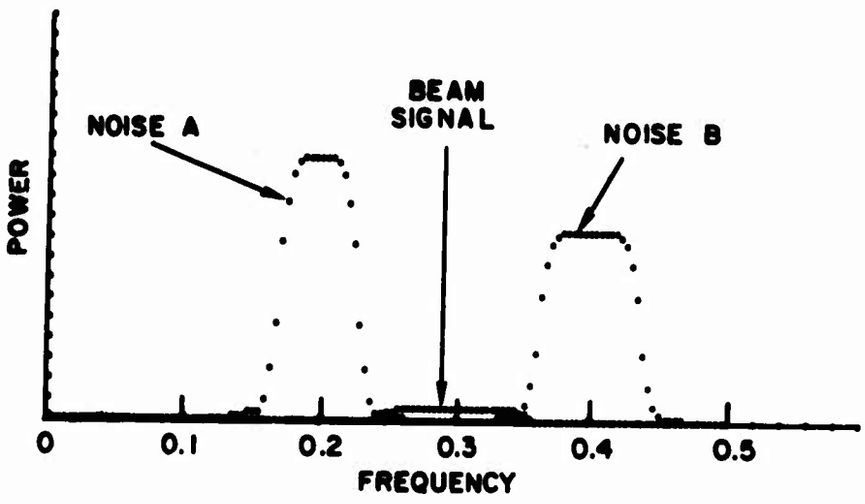


Fig. 6.5. Power spectral density of incoming signals.

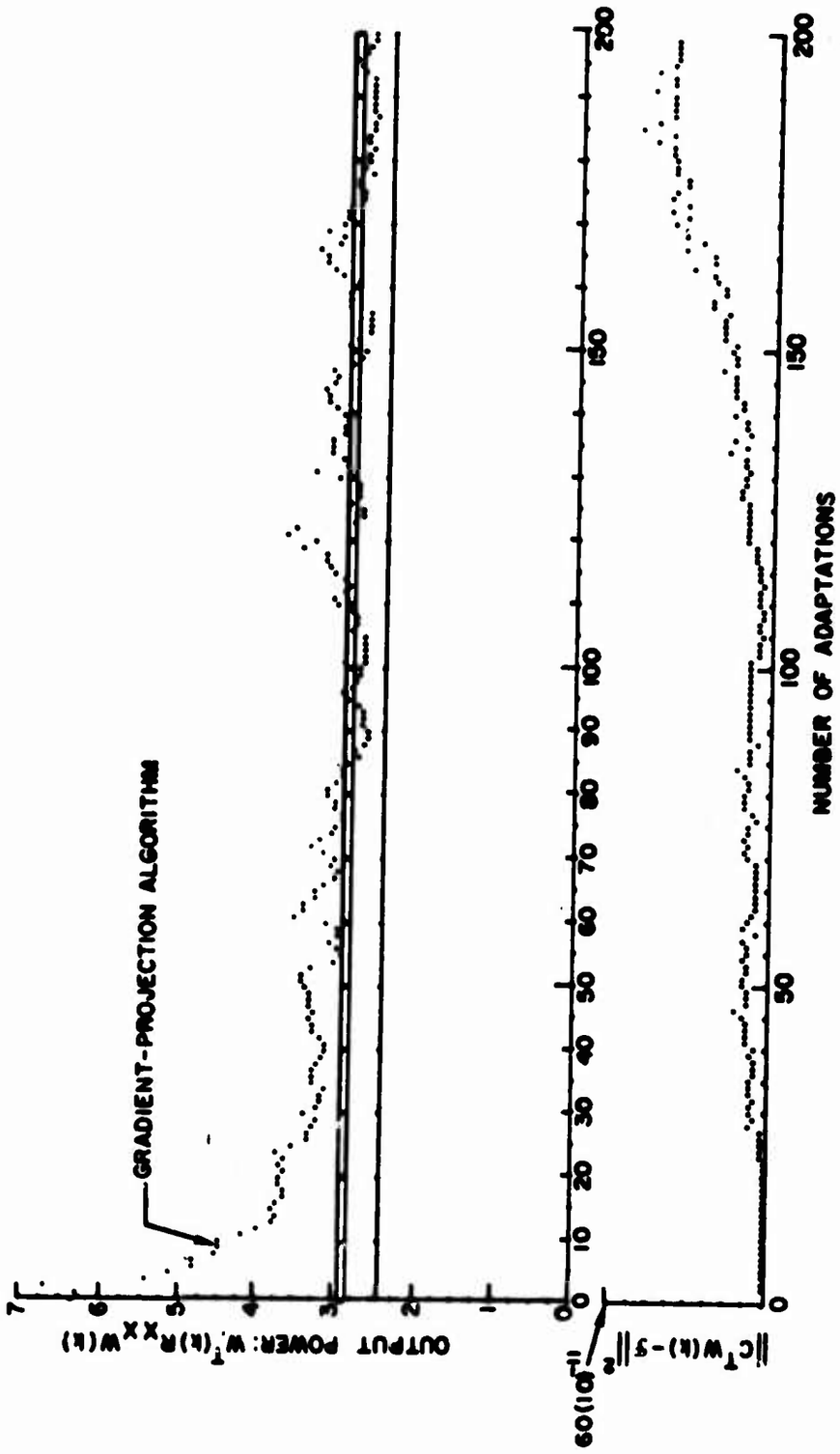


Fig. 6.6. THE OUTPUT POWER OF THE CONSTRAINED-LMS FILTER (upper graph) decreases as it adapts to discriminate against unwanted noise. Lower curve shows small deviations from the constraint due to quantization.

The processor was initialized with $W(0) = F = C(C^TC)^{-1}\mathcal{F}$, and Fig. 6.6 shows performance as a function of time. The upper graph has three horizontal lines. The lower line is the output power of the optimum weight vector. The closely-spaced upper two lines are upper and lower bounds for optimum output power plus misadjustment. The mean value of the processor's output power falls somewhere between the upper and lower bounds. The difference between the initial and steady state power levels is the amount of undesirable noise power the processor has been able to remove from the output.

Although the weight vector is, in theory, constrained to satisfy $C^TW(k) = \mathcal{F}$ at all times, very small deviations occur in an actual implementation due to quantization and computational errors. The lower graph in Fig. 6.6 shows the squared Euclidean distance between the weight vector and the constraint $\|C^TW(k) - \mathcal{F}\|^2$. An error-correcting feature of the Constrained-LMS algorithm prevents the deviations from the constraint from growing.

A last corollary deals with the deterministic constrained least squares problem.

Corollary 2.4. (Gradient-descent, Deterministic Constrained Least Squares) Let R_{XX} be a known $(n \times n)$ matrix and R_{Xd} be a known n-vector. The algorithm

$$W(k+1) = P[W(k) - \mu R_{XX} W(k) + \mu R_{Xd}] + F \qquad (6.33)$$

where

$$P = [I - C(C^T C)^{-1} C^T]$$

$$F = C(C^T C)^{-1} \mathfrak{f}$$

$$C^T W(0) = \mathfrak{f} \ ,$$

converges deterministically to the solution W_* of the problem

$$\begin{aligned} &\text{minimize} \quad [\alpha - 2W^T R_{Xd} + W^T R_{XX} W] \\ &\text{subject to} \quad C^T W = \mathfrak{f} \ , \end{aligned} \qquad (6.34)$$

where α is any finite constant, as long as μ satisfies (6.3). The convergence time along eigenvectors of $PR_{XX}P$ is given by (6.4) and there is no steady-state misadjustment.

Proof of Corollary 2.4.

This algorithm is the same as the recursive relation (5.2) for the mean weight vector of the stochastic constrained LMS algorithm. Showing that the stochastic algorithm

converges in the mean is therefore the same as showing (6.3) converges. Convergence in the mean was proved in Theorem 2.

This completes the proof of Corollary 2.4.

Remark: See Rosen [23] for an alternative solution to this problem.

VII. SENSITIVITY OF ALGORITHMS TO CALCULATION ERRORS

The constrained-LMS algorithm is related to the gradient-projection algorithms due to Rosen [23], Lacoss [14], and Booker [1]. The difference between the Constrained-LMS algorithm and gradient-projection algorithms lies in the way information about the location of the constraint surface is carried. As Fig. 4.5 showed, the Constrained LMS algorithm (3.11) "knows" the orientation of the constraint surface by the matrix C, and its translation from the origin by the vector F. In this section, it is shown that gradient-projection algorithms use only the orientation matrix C; to ensure that the weight vector stays on the constraint surface, they rely exclusively[†] on the fact that the weight vector is initialized on the constraint surface and always moves parallel to it. The gradient-projection method is shown to be sensitive to quantization errors which may cause the weight vector to deviate from the constraint on long runs.

Differences in the algorithms may be traced to Eq. (3.5) of the derivation. If $C^TW(k)$ is replaced by $\mathcal{F}$ in (3.5) and $R_{Xd} = \theta$, the gradient-projection algorithm of Booker results. ($C^TW(k)$ should equal $\mathcal{F}$ if $W(k)$ exactly satisfies

[†]Rosen recognized this problem and suggested using a second algorithm to "reset" the weight vector to the constraint whenever errors became excessive.

the constraint. It is shown in Fig. 6.6 that it may be unreasonable to assume that $W(k)$ is exactly on the constraint at all times. In the derivation of the Constrained-LMS algorithm, the term $C^TW(k)$ was carried instead of replacing it by $\mathcal{F}$. Carrying the term corresponds physically to assuming that $W(k)$ may not precisely satisfy the constraint, perhaps due to the quantization error of a digital implementation.

The algorithm that results from replacing $C^TW(k)$ by $\mathcal{F}$ is

$$W(k+1) = W(k) + \mu Pe(k)X(k) \; ; \quad C^TW(0) = \mathcal{F} \; . \tag{7.1}$$

This is a gradient-projection algorithm. It is so named because the unconstrained gradient is projected onto the constraint subspace and then added to the current weight vector. Its operation is shown in Fig. 7.1 (compare with Fig. 4.5).

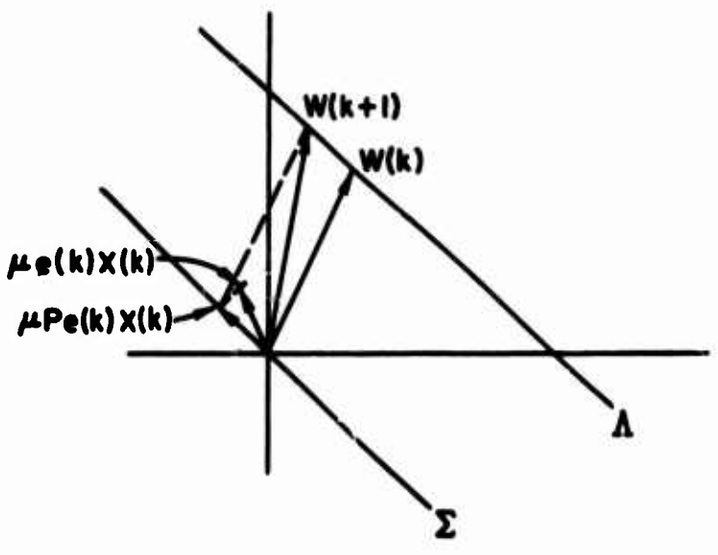


Fig. 7.1. Operation of the gradient-projection algorithm (7.1).

If somewhere in the computation an error occurs due to quantization and the weight vector is a bit off the constraint at time k, Fig. 7.2 shows that the Constrained-LMS algorithm (3.11) will bring the weight vector back to the constraint in the next iteration; however, the gradient-projection algorithm (7.1) assumes that $W(k)$ satisfied the constraint and adds a change parallel to the constraint surface, continuing the error.

An algebraic analysis is obtained by assuming that at each iteration the actual processor introduces a small vector of errors $\xi(k)$ to the weights. The update equations for the two algorithms become

Constrained-LMS:

$$\text{from (3.11):} \quad W(k+1) = P[W(k)+\mu e(k)X(k)] + F + \xi(k) \tag{7.2}$$

Gradient-Projection:

$$\text{from (7.1):} \quad W(k+1) = W(k)+\mu P e(k)X(k) + \xi(k) \; ;$$

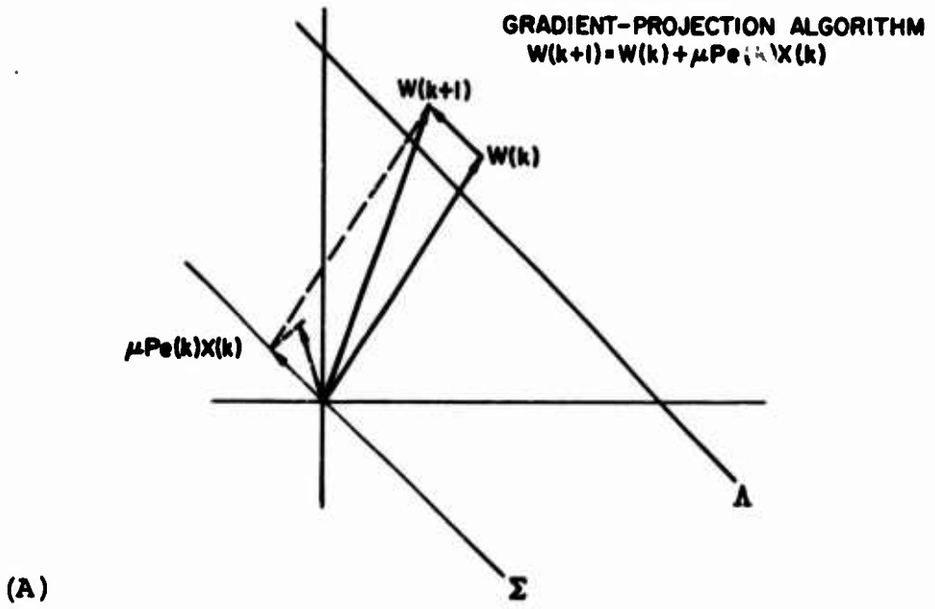
$$C^T W(0) = f \tag{7.3}$$

Iterating the Constrained-LMS algorithm (7.2) back to the original weight vector we have

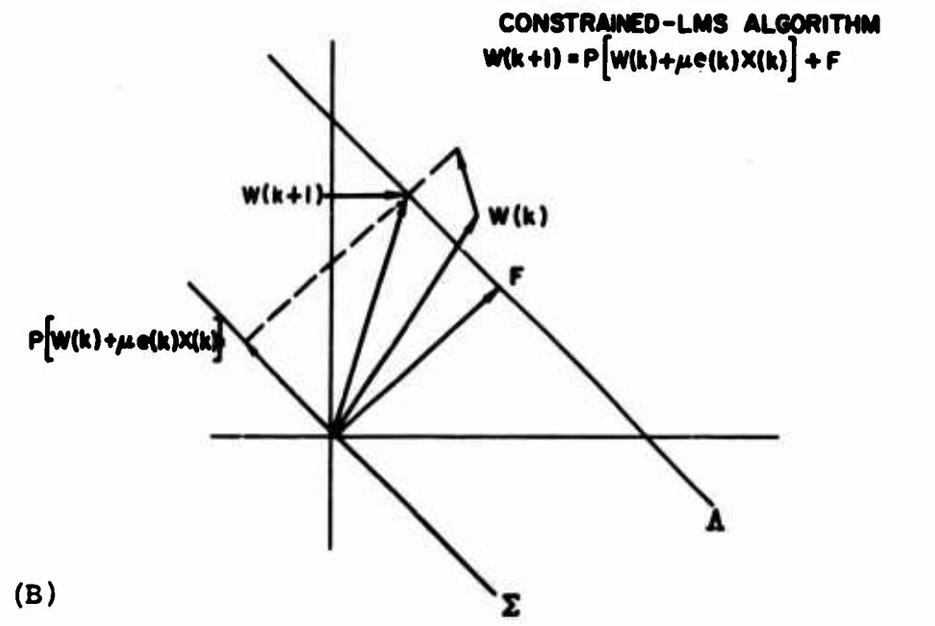
$$W(k+1) = P[W(k) - \mu X(k)X^T(k)W(0) + \mu X(k)d(k)] + F + \xi(k) \tag{7.4}$$

$$= \prod_{i=0}^{k} \{P[I-\mu X(i)X^T(i)]\}W(0)$$

$$+ \sum_{i=0}^{k} \prod_{j=i+1}^{k} \left\{P[I-\mu X(j)X^T(j)]\right\} [\mu P X(k)d(k) + F + \xi(i)] \tag{7.5}$$



(A)



(B)

Fig. 7.2. Error propagation. The Constrained-LMS algorithm (A) corrects deviations from the constraints while the gradient-projection algorithm (B) allows them to accumulate.

where undefined products are taken to be the identity. Now noting that $C^T P = C^T[I-C(C^TC)^{-1}C^T] = \theta$ and premultiplying by C^T to see how the weight satisfies the constraint we have

$$C^T W(k+1) = C^T[F+\xi(k)] = \mathcal{F} + C^T\xi(k) \quad . \tag{5.6}$$

In a perfect implementation the right side of (7.6) would be $\mathcal{F}$. With quantization errors and using the Constrained-LMS algorithm, the weight vector is off the constraint only by a term linear in the last error vector.

Now an error analysis on the gradient-projection algorithm is made. Performing a backward iteration on (7.3) produces

$$W(k+1) = W(k)-\mu P[X(k)X^T(k)W(k) - X(k)d(k)] + \xi(k) \tag{7.7}$$

$$= \prod_{i=0}^{k}\left\{I-\mu PX(i)X^T(i)\right\} W(0)$$

$$+ \sum_{i=0}^{k}\left\{\prod_{j=i+1}^{k}[I-\mu PX(j)X^T(j)]\right\}[\mu PX(i)d(i)+\xi(i)] \tag{7.8}$$

$$C^T W(k+1) = C^T W(0) + C^T \sum_{i=0}^{k} \xi(i) = \mathcal{F} + C^T \sum_{i=0}^{k} \xi(i) \tag{7.9}$$

The last term of (7.9) shows how the algorithm (7.1) accumulates deviations from the constraint.

If the computation errors are modeled as a zero-mean process [27], the gradient-projection algorithm does a

random walk, away from the constraint with variance increasing as the number of iterations (see Appendix D).

A simulation of the gradient projection algorithm on the array problem (Example 3) was made, using exactly the same data as used by the Constrained-LMS algorithm. The results are shown in Fig. 7.3. The lower part of Fig. 7.3 shows how the gradient-projection algorithm walks away from the constraint. Note the change in scale. If the errors of the Constrained LMS algorithm (Fig. 6.6) were plotted on the same scale they would not be discernible. Further, the errors of the gradient-projection method are expected to continue to grow.

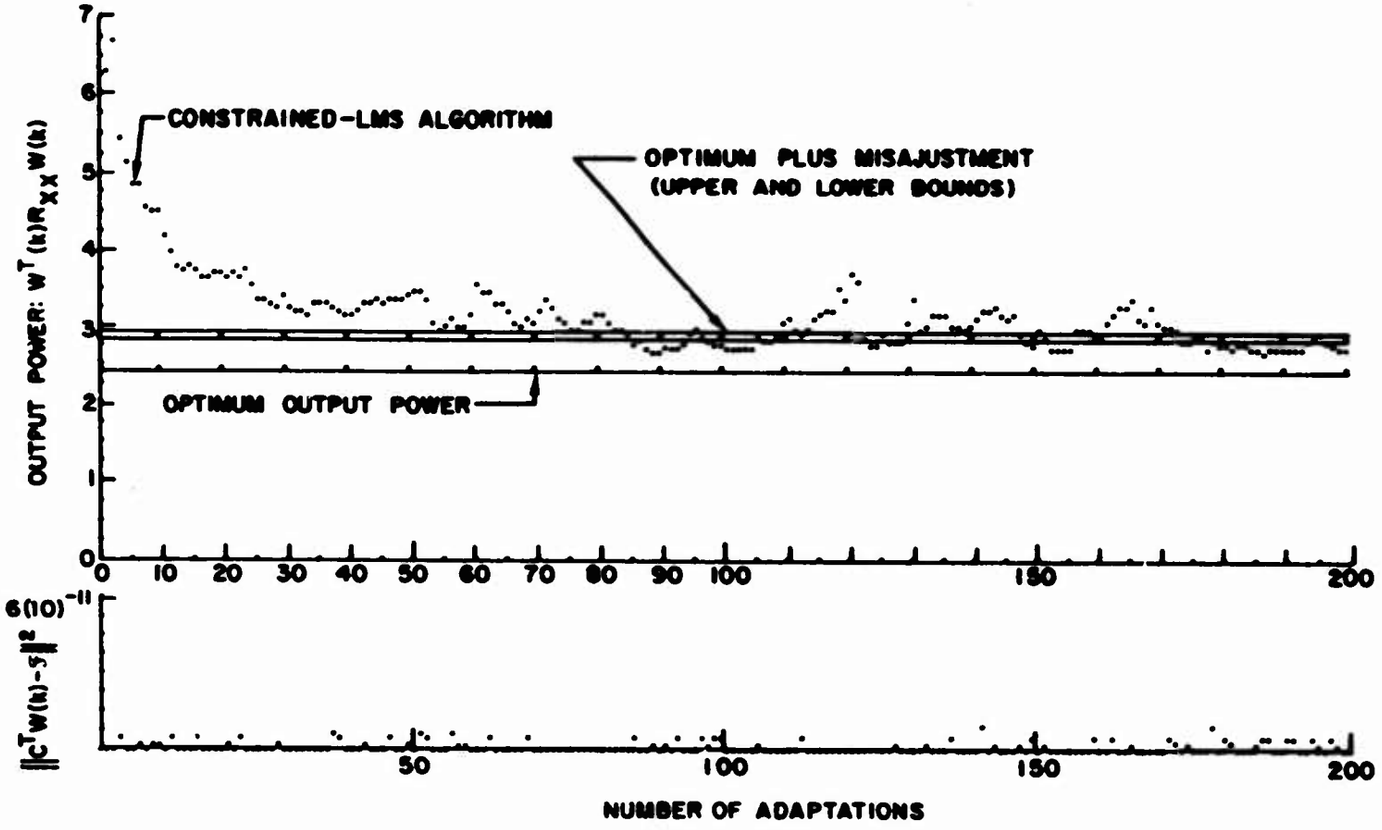


Fig. 7.3. OUTPUT POWER OF THE GRADIENT-PROJECTION ALGORITHM (upper graph) operated on the same data as the Constrained-LMS algorithm (c.f. Fig. 6.6). Lower curves shows that deviations from the constraint tend to increase with time. Note scale.

VIII. SUMMARY

A general algorithm was developed for stochastic linear least-squares optimization subject to linear equality constraints. The algorithm has three major properties: First, it has very modest computational requirements; second, it requires very little a priori knowledge; third, it converges to an optimal filter. A fourth property is that the algorithm can operate continuously without wandering from the constraints.

Rate of convergence and steady-state performance of the general algorithm are derived. Special cases of the algorithm are treated, with examples. An important application of the algorithm is the real-time processing of data from an array of sensors.

APPENDIX A

DERIVATION OF GRIFFITHS' MLR ALGORITHM BY THE QUADRATIC PENALTY FUNCTION METHOD

The purpose of this appendix is to show that the Maximum Likelihood Ratio (MLR) algorithm due to Griffiths [11] may be considered as an algorithm solving a least-mean-squares problem subject to "soft" linear equality constraints. This gives a simpler derivation than the original and immediately illuminates some properties of the algorithm that are well-known general properties of quadratic penalty function algorithms. As a side benefit, a general method of generating adaptive algorithms, based on the quadratic penalty function method, is indicated.

The quadratic penalty function method is a way of turning a constrained optimization problem into an easily-solved unconstrained optimization problem. Given a cost function $J(W)$ and a vector-valued constraint function $\Phi(W) = \theta$, the problem

$$\begin{aligned} &\text{minimize} \quad J(W) \\ &\text{subject to} \quad \Phi(W) = \theta \end{aligned} \tag{A.1}$$

is changed to

$$\text{minimize} \quad J(W) + \beta^2 \Phi^T(W)\Phi(W) \ . \tag{A.2}$$

As the scalar $\beta \to \infty$ the solution to the unconstrained problem (A.2) goes to the solution of the problem (A.1). The second problem is easily solved by standard unconstrained optimization techniques.

The specific problem considered by Griffiths is the problem of Example 3, Section VI, where $J(W) = W^T R_{XX} W$ and $\Phi(W) = C^T W - \mathcal{F}$. The algorithm is derived by forming the function

$$H(W) = \frac{1}{2} W^T R_{XX} W + \beta^2 (C^T W - \mathcal{F})^T (C^T W - \mathcal{F}) , \qquad (A.3)$$

and taking the gradient with respect to W

$$\nabla_W H = R_{XX} W + \beta^2 C (C^T W - \mathcal{F}) . \qquad (A.4)$$

The iteration is then

$$W(k+1) = W(k) - \mu \nabla_W H$$

$$= W(k) - \mu R_{XX} W(k) - \mu\beta^2 C (C^T W(k) - \mathcal{F}) . \qquad (A.5)$$

R_{XX} is replaced by its estimate, $X(k)X^T(k)$, giving

$$W(k+1) = W(k) - \mu X(k) X^T(k) W(k) - \mu\beta^2 C (C^T W(k) - \mathcal{F}) . \qquad (A.6)$$

This is Griffiths' MLR algorithm.

We infer from this derivation, and well-known properties of penalty function schemes [3], [17] that:

i) the algorithm has an error correcting property, i.e., it will not wander far from the constraint in the sense of the gradient projection algorithm discussed in Section VII.

ii) However, the satisfaction of the constraint is "soft", i.e., for finite values of β the solution of (A.2) will not exactly satisfy the constraint.

iii) Increasing β to cause the weight vector to more nearly satisfy the constraint will increase the convergence time of the algorithm.

APPENDIX B

STEADY-STATE MISADJUSTMENT

Moschner [20] calculated the misadjustment for the algorithm

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F \quad . \tag{B.1}$$

By his method precisely the same results for misadjustment may be obtained for the algorithm

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F \quad , \tag{B.2}$$

where $e(k) = d(k) - y(k)$ and the optimal weight vector of (B.2) is defined to be W_* of Theorem 1.

A slight improvement in the bounds obtained by Moschner is possible by noting in his equation (D.19) that since $B_n \triangleq E\{V_n V_n^T\}$ and $V_n = PV_n$ by Geometrical Property 5 and $P^2 = P$, then

$$\mathrm{Tr}(PRB_nR) = \mathrm{Tr}(PRPB_nR) \quad , \tag{B.3}$$

and so

$$\sigma_{min}\ \mathrm{Tr}(B_nR) \leq \mathrm{Tr}(PRB_nR) \leq \sigma_{max}\ \mathrm{Tr}(B_nR) \quad , \tag{B.4}$$

where σ_{min} and σ_{max} are the smallest and largest non-zero eigenvalues of PRP. The result follows by using the above facts in Moschner's derivation.

APPENDIX C

LEMMAS ON QUADRATIC FORMS

Lemma C.1. Let R be an $(n \times n)$ positive-definite matrix and C be an $(n \times m)$ matrix (with $n > m$) having full rank m. Then the $(m \times m)$ matrix $C^T R C$ is positive definite and $(C^T R C)^{-1}$ exists.

Proof of Lemma C.1.

Since R is positive definite then $V^T R V > 0$ for any n-vector $V \neq \theta$. We want to show for any m-vector $U \neq \theta$ that $U^T C^T R C U > 0$, hence, $C^T R C$ is positive definite and its inverse exists.

If the vector $U \neq \theta$, it has rank 1. By Sylvester's inequality [9], the rank of the product of two matrices is not less than the sum of the ranks of the matrices, less their common dimension. Letting $\rho(\cdot)$ denote rank, the rank of the n-vector CU is bounded by

$$\begin{aligned} \rho(CU) &\geq \rho(C) + \rho(U) - m \\ &\geq m + 1 - m \\ &\geq 1 , \end{aligned} \tag{C.1}$$

from which we conclude CU is not the zero vector. Therefore, letting $V = CU$ we conclude

$$U^T C^T R C U = V^T R V > 0 , \tag{C.2}$$

for any non-zero vector U so C^TRC is positive definite.

This completes the proof of Lemma C.1.

Remark 1. It follows that if R is positive definite R^{-1} is positive definite and $(C^TR^{-1}C)^{-1}$ exists.

Remark 2. Since the identity matrix is positive definite it follows that $(C^TC)^{-1}$ exists.

Lemma C.2. Let R be a positive-definite $(n \times n)$ matrix. Let $P = [I - C(C^TC)^{-1}C^T]$, where C is $(n \times m)$ with full rank m. Let the subspace Σ be defined as $\Sigma = \{W : C^TW = 0\}$. Then

i) m eigenvectors of PRP lie entirely outside Σ and have zero eigenvalues.

ii) The other $(n - m)$ eigenvectors of PRP lie entirely within Σ and have strictly non-zero eigenvalues.

iii) Let σ_i be the i^{th} non-zero eigenvalue of PRP and λ_j be the j^{th} eigenvalue of R. Then the eigenvalues are related by

$$\lambda_{min} \leq \sigma_{min} \leq \sigma_i \leq \sigma_{max} \leq \lambda_{max}, \qquad (C.3)$$

for all $i = 1, 2, \ldots (n-m)$.

Proof of Lemma C.2.

Since PRP is a symmetric $(n \times n)$ matrix, it has n eigenvectors and n eigenvalues. The eigenvectors can be chosen to be orthogonal [7].

i) Since the matrix C has full rank it has m columns of linearly independent n-vectors. Direct calculation shows that $C^T PRP = \theta$, so the m columns of C are eigenvectors of PRP with zero eigenvalues.

ii) There must be $(n-m)$ remaining eigenvectors orthogonal to the columns of C. As shown in Appendix E, the columns of C are vectors normal to the constraint plane Γ and subspace Σ. Therefore, the remaining $(n-m)$ eigenvectors must be in Σ. As shown in Geometrical Property 5 of Section IV, if V is a vector in Σ, then $PV = V$. Therefore if an eigenvector e_i of PRP is in Σ then

$$e_i^T PRPe_i = e_i^T Re_i > 0 . \qquad (C.4)$$

Let σ_i be an eigenvalue corresponding to an eigenvector of PRP in Σ. Then by definition

$$PRPe_i = \sigma_i e_i \qquad (C.5)$$

so

$$e_i^T PRPe_i = \sigma_i e_i^T e_i = \sigma_i . \qquad (C.6)$$

From (C.4) and (C.6) it follows that

$$\sigma_i > 0 \qquad i=1,2,\ldots,(n-m) \ . \tag{C.7}$$

iii) It is well known that if e is a unit vector then $e^T R e$ is bounded by

$$\lambda_{min} \leq e^T R e \leq \lambda_{max} \ , \tag{C.7}$$

where λ_{min} and λ_{max} are respectively the largest and smallest eigenvalues of R . Therefore from (C.4) and (C.6)

$$\lambda_{min} \leq \sigma_i \leq \lambda_{max} \ . \tag{C.8}$$

The result follows.

<u>This completes the proof of Lemma C.2.</u>

APPENDIX D

EXPECTED DEVIATION FROM THE CONSTRAINT BY THE GRADIENT-PROJECTION ALGORITHM

As an approximation, quantization in the weight vector is modeled as an additive white noise process (see Widrow, [27]); the expected deviation from the constraint by the gradient projection algorithm is computed as a function of time.

Assume that a fixed-point representation for the weights is used; let the quantization size of a single weight be q. Using Widrow's value for the error variance, $q^2/12$, from (7.9) the expected squared Euclidean distance from the constraint at time k is

$$E\{\|C^T W(k) - \mathcal{F}\|^2\} = E\{\sum_{i=1}^{k} \xi_i^T C C^T \sum_{j=1}^{k} \xi_j\}$$

$$= \mathrm{Tr}(E\{C \sum_{i=1}^{k} \sum_{j=1}^{k} \xi_i \xi_i^T\} C^T)$$

$$= \mathrm{Tr}(C \frac{kq^2}{12} I C^T)$$

$$= \frac{kq^2}{12} \mathrm{Tr}(CC^T) \qquad \text{(D.1)}$$

Thus the expected squared distance from the constraint increases linearly with time (approximately).

For the special case of the array problem, with C defined by Eq. (6.27), $\mathrm{Tr}(CC^T) = n$, where n is the number of tap points. Equation (D.1) becomes

$$E\{\|C^T W(k) - \mathcal{F}\|^2\} = kn \frac{q^2}{12} . \qquad \text{(D.2)}$$

APPENDIX E

THE METHOD OF LAGRANGE MULTIPLIERS

Consider the equality-constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad J(W) \\ &\text{subject to} \quad \Phi(W) = \theta \end{aligned} \tag{E.1}$$

where $J(\cdot)$ is a scalar cost function and $\Phi(\cdot)$ is a vector-valued constraint function. In Theorem 1, $J(W) = E\{(d - W^T X)^2\}$ and $\Phi(W) = C^T W - \mathcal{F}$. Let the gradient of the function $J(W)$ with respect to a vector W evaluated at W_o be written $\nabla_W J(W_o)$ where

$$\nabla_W J(W_o) = \left[\begin{array}{c} \frac{\partial J}{\partial w_1} \\ \frac{\partial J}{\partial w_2} \\ \vdots \\ \frac{\partial J}{\partial w_m} \end{array} \right]_{W = W_o} . \tag{E.2}$$

A necessary requirement for the optimal solution of (E.1) to be at a point W_o is that the gradient of J with respect to W be normal to the constraint surface† at W_o. If the gradient of J at W_o were not normal to the constraint surface then by sliding along the constraint a vector W_1 could be found still satisfying the constraint but having

†The constraint surface is understood to be the points satisfying the constraint $\Phi(W) = \theta$.

lower cost, i.e., $J(W_1) < J(W_o)$.

Fleming ([8], p. 126) shows that the normal vectors to any manifold defined by $\Phi(W) = \theta$ is $\nabla_W \Phi$. For example, the gradient of the constraint defined by $\Phi(W) = C^T W - \mathcal{F} = \theta$ is $\nabla_W (C^T W - \mathcal{F}) = C$; therefore, each of the m columns of C is a vector orthogonal to the constraint plane and any linear combination of those vectors is also orthogonal to the plane.

Let λ be an undetermined m-vector of multipliers. The vector $C\lambda$ is a linear combination of the columns of C and so is normal to the constraint plane. Thus another way to express the necessary condition that the gradient of the cost function J be orthogonal to the constraint surface is to say that for some choice of λ the gradient and the normal may be anticollinear, i.e., (see Fig. E.1)

$$\nabla_W J(W_o) + C\lambda = \theta \ , \tag{E.3}$$

or more generally

$$\nabla_W J(W_o) + \nabla_W \Phi(W_o)\lambda = \theta \ . \tag{E.4}$$

Another way of writing (E.4) analogous to the necessary condition for unconstrained optimality ($\nabla_W J(W_o) = \theta$) is by defining the function $H(W)$ by "adjoining" the cost function to the constraint function by the Lagrange multipliers,

$$H(W) = J(W) + \lambda^T \Phi(W) \tag{E.5}$$

and requiring

$$\nabla_W H(W) = \theta \ . \tag{E.6}$$

Then since $\nabla_W H(W_o) = \nabla_W H(W_o) = \nabla_W J(W_o) + \nabla_W \Phi(W_o)\lambda$, (E.6) is identical to (E.4) and the necessary conditions become (E.6) and

$$\Phi(W_o) = \theta \ . \qquad \text{(E.7)}$$

For an excellent discussion of the Lagrange multiplier method in more general applications see Bryson and Ho [3].

APPENDIX F

SIMULATION OF THE DIRECT SUBSTITUTION ALGORITHM

At the beginning of Section III the direct substitution algorithm was suggested: To obtain an estimate of the optimal weight vector, the unknown correlation matrices are estimated and inserted directly into the equation for the optimal weight vector. Although computationally quite difficult (because of the number of matrix inversions and multiplications involved) the direct substitution method offers the possibility of improved performance.

The direct substitution algorithm was simulated on the array-processing problem of Example 3 using exactly the same data as the Constrained-LMS processor. The direct substitution algorithm is

$$\hat{R}_{XX}(k) = \alpha \hat{R}_{XX}(k-1) + (1-\alpha)X(k-1)X^T(k-1) \qquad \text{(F.1)}$$

$$W(k) = \hat{R}_{XX}^{-1}(k)C[C^T\hat{R}_{XX}^{-1}(k)C]^{-1}\mathcal{F} \qquad \text{(F.2)}$$

where $0 < \alpha < 1$. Equation (F.1) is an exponentially-weighted estimate of the true correlation matrix R_{XX} . Equation (F.2) is the equation for the optimal weight vector for the problem with $\hat{R}_{XX}(k)$ substituted for R_{XX} . The constant α , which controls both rate of convergence and misadjustment, was chosen to be 0.97, a value which experimentally lead to approximately the same misadjustment as the Constrained-LMS processor had in Example 3. $\hat{R}_{XX}(0)$ was initialized to the

identity matrix, scaled by the power measured on each tap; in this case the total power on each tap was 2.2 (see Table 6.1), so $\hat{R}_{XX}(0)$ was $2.2\mathbf{I}$. This is a reasonable starting point since the power on a tap is easily measured in a real situation and also a simple calculation shows that if $\hat{R}_{XX}(0)$ is any diagonal matrix then $W(0) = C(C^TC)^{-1}g = F$. The vector F was also the initial weight vector of the Constrained-LMS algorithm so the two processors essentially start out at the same point and a meaningful comparison is easily obtained.

Results of the simulation are shown in Fig. F.1. Compare with Fig. 6.6. For the same misadjustment, the better processor should have a faster rate of convergence. A careful comparison of Fig. F.1 and 6.6 fails to show conclusively which algorithm has the better performance. For this example at least, the user would have been just as well off to use the simpler Constrained-LMS processor.

Readers interested in the direct substitution method should consult Saradis, et al. [24] and Mantey and Griffiths [18], [19].

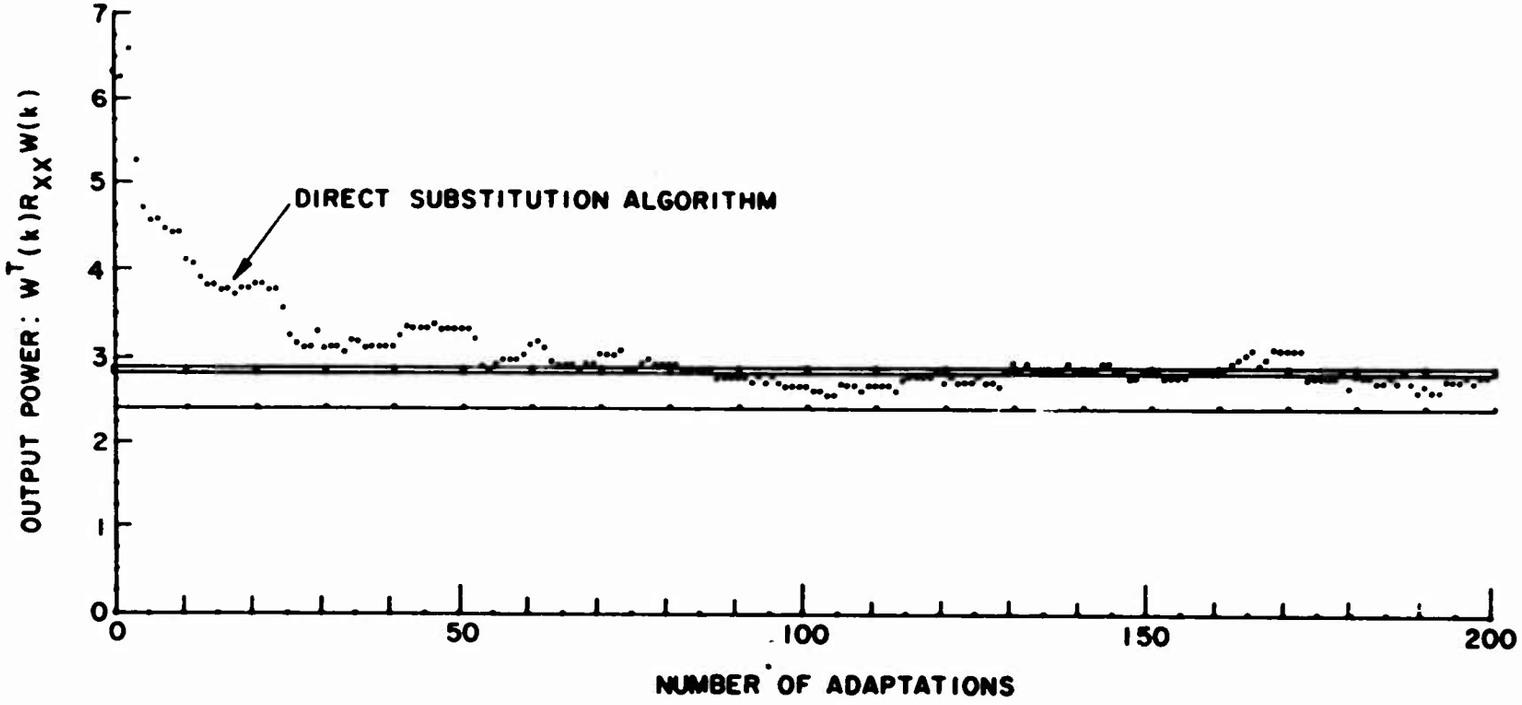


Fig. F.1. Performance of the direct substitution algorithm on the same data used by the Constrained-LMS algorithm (Fig. 6.6). The horizontal lines are retained for reference.

BIBLIOGRAPHY

1. A. Booker, et al., "Multiple-constraint adaptive filtering", Texas Instruments, Science Services Division, Dallas, Texas, Apr 1969

2. J. E. Brown, III, "Adaptive estimation in nonstationary environments", Ph.D. dissertation in preparation, Stanford Electronics Laboratories, Stanford, Calif.

3. A. E. Bryson, Jr. and Y. C. Ho, Applied Optimal Control, Blaisdell Publishing Co., Waltham, Mass., 1969

4. J. Capon, et al., "Multidimensional maximum-likelihood processing of a large aperture seismic array", Proc. IEEE, 55, Feb 1967, pp. 142-211.

5. R. E. Collin and F. J. Zucker, Antenna Theory, Part I, McGraw-Hill, New York, 1969

6. T. P. Daniell, "Adaptive estimation with mutually correlated training samples", SEL-68-083 (TR No. 6778-4), Stanford Electronics Laboratories, Stanford, Calif., Aug 1968

7. P. M. DeRusso, et al., State Variables for Engineers, John Wiley & Sons, New York, 1965

8. W. H. Fleming, Functions of Several Variables, Addison-Wesley, Reading, Mass., 1965.

9. F. R. Gantmacher, The Theory of Matrices, Chelsea Press, New York, 1959.

10. I. J. Good and K. Doog, "A paradox concerning rate of information", Information and Control, 1, 2, May 1958, pp. 113-126.

11. L. J. Griffiths, "Signal extraction using real-time adaptation of a linear multichannel filter", SEL 68-017, (TR No. 6788-1), Stanford Electronics Laboratories, Stanford, Calif., Feb 1968

12. L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays", Proc. IEEE, 57, 10, Oct 1969, p. 1696.

13. R. Kneipfer and R. Hilt, "System description of NUSL's iterative adaptive beamformer (ITAB)", NUSL Tech. Memo. No. 2242-63-70, Naval Undersea Sound Laboratory, Fort Trumbull, New London, Conn., Mar 1970

14. R. T. Lacoss, "Adaptive combining of wideband array data for optimal reception", IEEE Trans. Geoscience Elect., GE-6, 2, May 1968

15. A. Lender, "Decision-directed adaptive equalization technique for high-speed data transmission", Proc 1970 Intl. Conf. on Communications, San Francisco, Jun 1970

16. David G. Luenberger, Optimization by Vector State Methods, John Wiley & Sons, New York, 1969

17. D. G. Luenberger, "Convergence rate of a penalty function scheme", Internal Memo. 69-1, Dept. of Engr-Econ. Systems, Stanford University, Stanford, California

18. P. E. Mantey and L. J. Griffiths, "Iterative least-squares algorithms for signal extraction", Proc. 2nd Hawaii Intl. Conf. on Syst. Sci., pp. 767-770.

19. P. E. Mantey and L. J. Griffiths, Manuscript in preparation.

20. J. L. Moschner, "Adaptive filtering with clipped input data", Ph.D. dissertation, Stanford Electronics Laboratories, Stanford, Calif., Jun 1970

21. A. H. Nuttall, "Theory and application of the separable class of random processes", TR 343, Res. Lab.of Electronics, M.I.T., Cambridge, Mass., 1958

22. A. H. Nuttall and D. W. Hyde, "A unified approach to optimum and suboptimum processing for arrays," USL Rept. No. 992, U. S. Naval Undersea Sound Laboratory, Fort Trumbull, New London, Conn., Apr 1969

23. J. B. Rosen, "The gradient projection method for nonlinear programming, pt. I: Linear constraints", J. Soc. Indust. Appl. Math., 8, 1, Mar 1960, p. 181

24. G. N. Saradis, et al., "Stochastic approximation algorithms for system identification, estimation, and decomposition of mixtures", IEEE Trans. of Syst. Sci. and Cyber., SSC-5, 1, Jan 1969

25. J. L. Steinberg and J. Lequeux, Radio Astronomy, McGraw-Hill, New York (translated by R. N. Bracewell)

26. H. L. Van Trees, Detection, Estimation, and Modulation Theory, Part I, John Wiley & Sons, New York, 1968

27. B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory", IRE Trans. Professional Group on Circuit Theory, CT-3, 4, Dec 1956

28. B. Widrow, "Adaptive filters I: fundamentals", SEL-66-126 (TR No. 6764-6), Stanford Electronics Laboratories, Stanford, Calif., Dec 1966

29. B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits", IRE WESCON Conv. Rec., Part 4, pp. 96-104, 1960

30. B. Widrow, et al , "Adaptive antenna systems", Proc. IEEE, 55, 12, Dec 1967

PART II

ADAPTIVE ESTIMATION IN NONSTATIONARY ENVIRONMENTS

by

James Edward Brown, III

ABSTRACT

In the classical design of processors for sensor arrays whose purpose is signal detection and estimation, a receiver is optimized on the basis of the a priori knowledge of the statistics of its input signals. However, when the a priori knowledge is not available, the receiver's performance can still be improved by performing measurements on its input signals and incorporating this new information into its design. Such receivers are called adaptive.

The purpose of this research is to develop and analyze a gradient-descent surface-searching algorithm for automatically adjusting (adapting) the parameters of a linear tapped-delay-line array processor in order to improve its performance in an unknown changing environment. The tracking ability of this algorithm is demonstrated when the characteristics of the nonstationarity are such that the optimum parameter sequence can be modeled as a first-order Markov process with a known transition function. A worst-case analysis of the algorithm is presented for three types of nonstationarities when the above model for the nonstationarity is not applicable.

The techniques developed in analyzing the above algorithm provide a powerful approach for the further study of gradient-descent algorithms used in searching unknown, nonstationary surfaces. Among the most important consequences are:

i) the removal of the usual assumption that the data be jointly Gaussian;

ii) the development of a new convergence theorem for a dynamic stochastic approximation algorithm, thereby extending a branch of stochastic approximation theory to the analysis of adaptive processors in nonstationary statistics;

iii) the enlargement of the class of problems for which stochastic approximation algorithms, adaptive estimation algorithms, and the Kalman-Bucy theory can be compared.

Also presented in an appendix is a procedure for automatically adjusting the convergence factor. Some experimental results are presented.

ACKNOWLEDGMENT

I am grateful to Drs. Therill Valentine and Aaron Booker of Texas Instruments, Inc., for introducing me to the field of adaptive estimation. The stimulating discussions I had with them cultivated my interest in this area.

I wish to thank Drs. Bernard Widrow and Thomas M. Cover for their guidance and encouragement given throughout this research. Appreciation is also expressed to Dr. Gerald Pearson for his reading of this report.

I am particularly grateful for the many helpful discussions I have had with Dr. Thomas P. Daniell and my colleagues Michel Installé and Otis L. Frost in formulating my idea.

Special appreciation goes to Miss Barbara Kenyon for the very difficult job of typing this manuscript.

The principal support for this research was a NASA Traineeship. Additional support was given by Naval Ship Systems Command Contract No. N00024-69-C-1430 and the Naval Underwater Sound Laboratory Contract No. N00140-69-C-0341.

TABLE OF CONTENTS (PART II)

LIST OF ILLUSTRATIONS

SYMBOLS

B	bound for optimum weight-vector sequence
$B(\mu)$	bound coefficient as a function of μ
b_n^2	expected norm-squared difference between optimum weight-vector and adaptive weight-vector at time n
C_n^2	excess mean-squared-error at time n
c_n^2	expected norm-squared difference between a fixed weight-vector and an adaptive weight-vector at time n
d_n	desired output of array processor at time n
$E[\cdot]$	expected-value operator
E^p	Euclidean p-space
$F_n(\cdot)$	evolution operator for the optimum weight-vector process at time n
f_n	Lipschitz coefficient for F_n
f	limit superior of the sequence $\{f_n\}$
$G_n(\cdot)$	evolution operator for the optimum weight-vector process at time n
g_n	Lipschitz coefficient for G_n
g	limit superior of the sequence $\{g_n\}$
I	identity matrix
$J_n(\cdot)$	gradient of the mean-square-error surface at time n
LMS	least-mean-square
P_n	crosscorrelation vector at time n
p	dimension of vector space
R_n	correlation matrix at time n
$(\cdot)^T$	matrix-transpose operator
U_n	driving random vector for the optimum weight-vector

W	p-dimensional column weight vector
W_o^*	fixed weight-vector
W_n^*	optimum p-dimensional linear estimator at time n
W_n	adaptive algorithm estimate of W_n^*
w_i	i^{th} component of W
X	p-dimensional array input vector
X_n	p-dimensional array input vector at time n
x_i	i^{th} component of X
y_n	output of array processor at time n
z_n	estimate of $J_n(W_n)$ at time n
α	a bound coefficient
$\alpha(\mu)$	a bound coefficient as a function of μ
β	a bound coefficient
$\beta(\mu)$	a bound coefficient as a function of μ
Δ_k	norm-squared change in optimum weight-vector at time k
Δ	limit superior of the sequence $\{E[\Delta_k^2]\}$.
$\Gamma_1(\varepsilon)$	bound parameter as a function of ε
$\Gamma_2(\varepsilon)$	bound parameter as a function of ε
δ	a tolerance
ϵ	bound parameter
ϵ_0	bound parameter
λ^*	upper-bound for the eigenvalues of R_n
λ_*	lower-bound for eigenvalues of R_n
$\lambda_{max}(x)$	maximum eigenvalue of R_n
$\lambda_{min}(x)$	minimum eigenvalue of R_n

μ_n	gain parameter for adaptation algorithm
$\xi_n(\cdot)$	mean-squared-error function at time n
ξ_n^*	minimum mean-squared-error at time n
$\Pi(\cdot)$	product of
ρ_n	second moment of U_n
ρ	limit superior of the sequence $\{\rho_n\}$
σ_1	bound coefficient
σ_2	bound coefficient
$\Sigma(\cdot)$	summation of
$\{\cdot\}$	sequence
$\triangleq$	defined as
$\lvert\cdot\rvert$	absolute value of
$\lVert\cdot\rVert$	norm of
$\exp(\cdot)$	exponential function
$\lim_{n\to\infty}(\cdot)$	limit of the sequence $\{\cdot\}$
$\limsup_{n\to\infty}(\cdot)$	limit superior of the sequence $\{\cdot\}$
$\sup_n(\cdot)$	least upper bound for the sequence $\{\cdot\}$
$\inf_n(\cdot)$	greatest lower bound for the sequence $\{\cdot\}$
$\longrightarrow$	converges to
$\searrow$	converges monotonically down to
$\nearrow$	converges monotonically up to

I. INTRODUCTION

A. PURPOSE

In the classical design of processors for sensor arrays whose purpose is signal detection and estimation [1]-[7], a crucial role is played by the a priori information available at the receiver. In practice, the receiver's performance can be improved by performing measurements on its input signals and incorporating this new information into its design [11]-[21]. Such receivers are called adaptive.

The purpose of this research is to develop and analyze a procedure for automatically adjusting (adapting) the processor in order to improve its performance in an unknown changing environment.

B. THE PROBLEM

The type of array processor considered in this paper is the multichannel linear discrete time processor (also referred to as the tapped-delay-line processor) shown in Fig. 1.1. The input x at each receiver is sampled at regular intervals and shifted down the tapped delay line. The sampled value at each tap is weighted, and all the weighted values are summed to form an output y which will be viewed as an estimate of some desired quantity d. For the simple case when x consists of a transmitted signal plus additive noise, the desired output d is taken to be the transmitted signal. In general, d may be taken to be

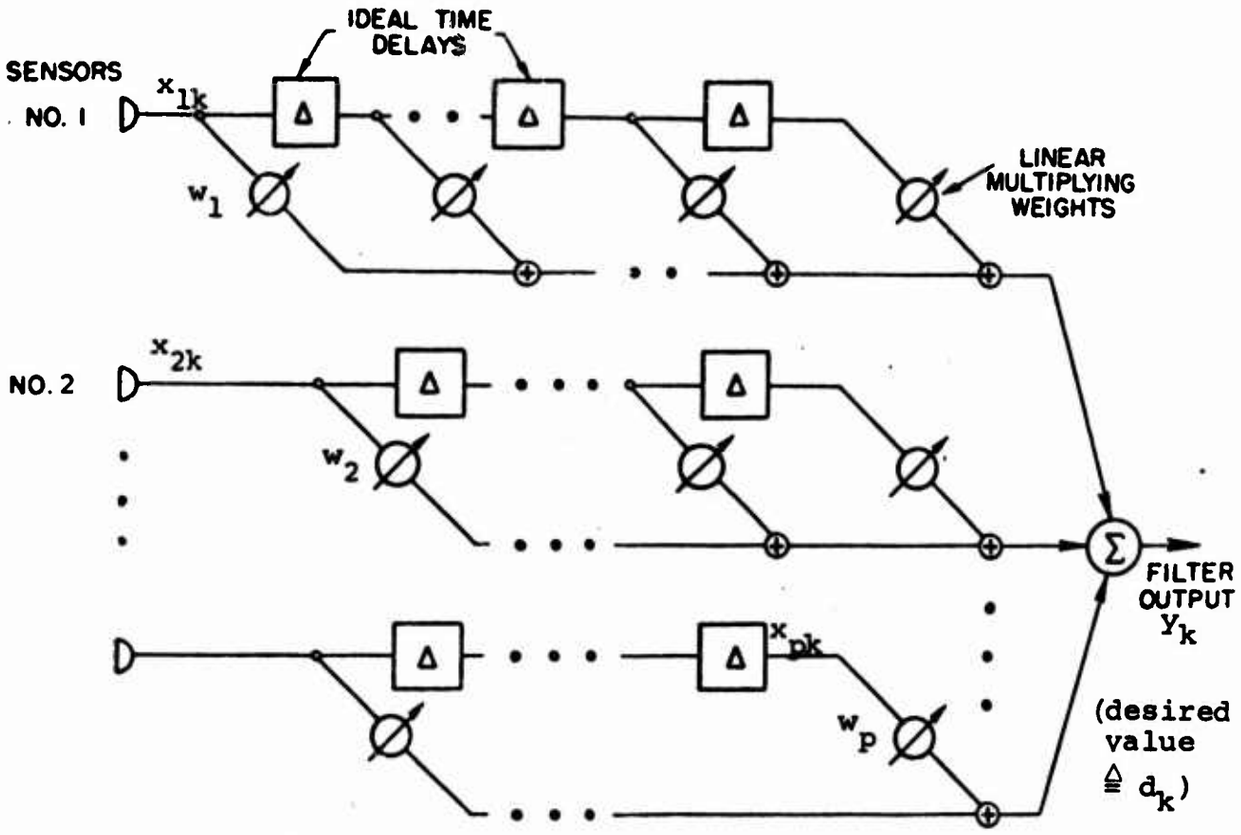


Fig. 1.1. General form of tapped-delay-line multichannel filter.

some other desired output, depending on the purpose of the receiver.

The criterion used in this research for determining the best set of weight values w for the above system is the mean-squared error between the processor output y and the desired output d [1] - [7]. This criterion is a common one used in the design of array processors since the pioneering work of Wiener [4].

The mathematical description of the array problem is given as follows: Let $\{X_k : k = 1,2,\ldots\}$ be a sequence of p-dimensional vector-valued random variables to be referred to as the input sequence. The components of X_k are the inputs $x_{1k}, x_{2k}, \ldots, x_{pk}$ at the various taps of the processor at time k. Let $\{d_k : k = 1,2,\ldots\}$ be a corresponding sequence of real-valued random variables to be referred to as the desired-output sequence. The pair $(d_k, X_k,)$ will be called the data pair at time k. Assume that the sequence $\{(d_k, X_k) : k = 1,2,\ldots\}$ is an independent sequence. The correlation matrix at time $k = n$, defined by

$$R_n \triangleq E[X_n X_n^T]^{\dagger}, \qquad (1.1)$$

is assumed to be positive definite with finite eigenvalues. The crosscorrelation vector at time $k = n$ is defined by

---

†The expectation operator will be denoted by $E[\cdot]$. The transpose will be denoted by T.

$$P_n \triangleq E[d_n X_n] \quad . \tag{1.2}$$

Let W be some p-dimensional column vector, referred to as the weight vector or discrete-time filter, whose components are the weights $w_1, w_2, \ldots, w_p$.

The object is to estimate that weight vector W_n^* which minimizes the mean-squared error at time n, given by

$$\xi_n(W) = E[(d_n - W^T X_n)^2]$$

$$= E[d_n^2] - 2W^T P_n + W^T R_n W \quad . \tag{1.3}$$

It is a well known result [42] that W_n^* is given by

$$W_n^* = R_n^{-1} P_n \quad . \tag{1.4}$$

This vector, W_n^*, will be called the "Wiener" weight vector or the optimum finite-dimensional linear weight vector at time n. The corresponding mean-squared error will be denoted by ξ_n^*. (A typical mean-squared-error surface is shown in Fig. 1.2 when the weight vector has only two components.) Note that the expression (1.4) requires that the second order statistics, R_n and P_n, be known. It would be highly desirable to have a design procedure which would not require this a priori information since it normally would not be available to the array processor.

A method for determining R_n and P_n that immediately comes to mind is to compute the time-averages [14]

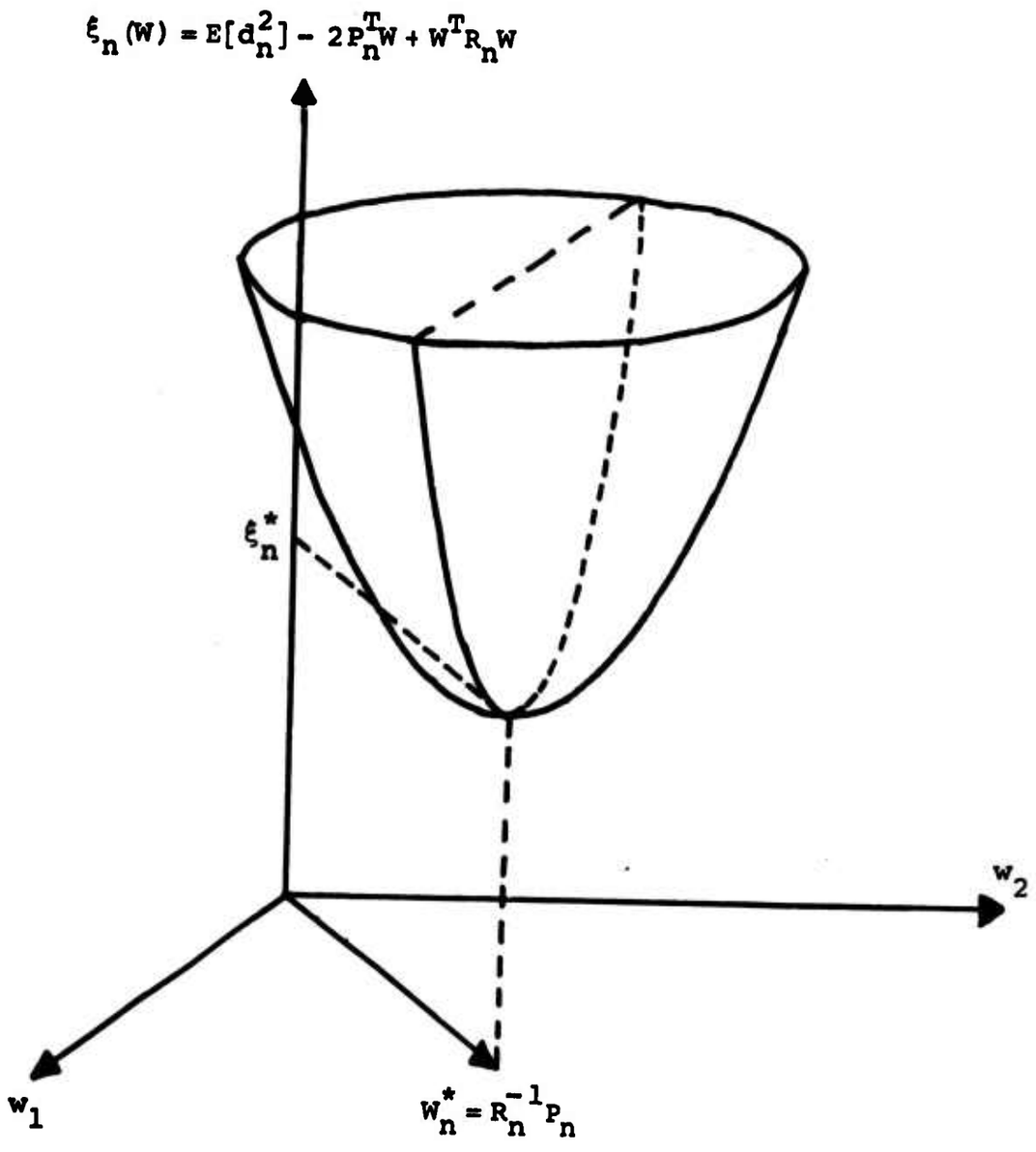


Fig. 1.2. A Two-dimensional Quadratic Mean-squared-error Surface

$$\frac{1}{n}\sum_{k=1}^{n} X_k X_k^T \quad \text{and} \quad \frac{1}{n}\sum_{k=1}^{n} d_k X_k \; . \tag{1.5}$$

For the stationary problem this would result in the optimum finite-dimensional linear estimator W^* in the limit as more and more measurements become available. However, if the environment in which the receiver operates is nonstationary, the above method is not applicable in the determination of the instantaneous value of the optimum weight vector W_n^*. The time-averages (1.5) progressively weighs the new information contained in the data pair (d,X) less and less as time progresses. Meanwhile, the optimum weight vector continues to change. Another procedure for estimating W_n^* will have to be developed.

C. APPROACH

The approach used in this research for estimating (or tracking) the optimum weight-vector sequence $\{W_n^*\}$ is to extend a gradient-descent surface-searching algorithm (the method of steepest descent [8] - [10]) to the tracking of an unknown time-varying surface. The resulting system (array processor plus adaptation algorithm) gains the capability of responding to changes in the input-data statistics. This results in an adaptive system whose performance is vastly superior to that of a fixed system in many instances.

The analysis of the adaptation algorithm is divided into two parts. In Chapter IV asymptotic bounds for the performance of an adaptive processor are obtained when its input is

nonstationary. The characteristics of the nonstationarity are such that the optimum weight-vector sequence $\{W_n^*\}$ can be modeled as a first-order Markov process with a known transition function. However, in many applications it is unreasonable to expect that a model for the nonstationarity will be available. For this reason a worst-case analysis of the adaptive system is presented in Chapter V for three types of nonstationarities. As shown by the example given in Chapter VI, these results are particularly informative as to the type of behavior to expect from the adaptive system.

D. CONTRIBUTIONS

The principle contributions of this research are:

1) A gradient-descent surface-searching algorithm is developed for adapting the parameters of a linear tapped-delay-line array processor in an unknown, time-varying environment. The tracking ability of this algorithm is demonstrated when the nonstationarity is modeled by the optimum weight-vector sequence $\{W_n^*\}$ being a first-order Markov process with a known transition function. A worst-case analysis of the algorithm is presented for three types of nonstationarities when the above model for the nonstationarity is not applicable.

2) The techniques developed in analyzing the above algorithm provide a very powerful approach for the further study of gradient-descent algorithms used in searching unknown,

nonstationary surfaces. Among the most important consequences are:

(i) the removal of the usual assumption that the data pair (d,X) be jointly Gaussian.

(ii) the development of a new convergence theorem for a dynamic stochastic approximation algorithm, thereby extending a branch of stochastic approximation theory to the analysis of adaptive array processors in nonstationary statistics.

(iii) the development of an analytical comparison between the adaptation algorithm and the Kalman-Buey recursive filter. This result is presented in Appendix I.

II. TRADITIONAL METHODS FOR DESIGNING ADAPTIVE PROCESSORS

A large number of array processors which adjust their parameters as a function of their inputs have been considered in the literature. A representative sample is given in the references [11] - [52]. Two basic types of systems have resulted from the above research: a parametric system [11] - [13] and a non-parametric system [14] - [52]. The parametric system is characterized by the assumption of an underlying statistical framework for the input data; e.g., (d,X) jointly Gaussian, or the waveform of d known with X Gaussian, etc. This system is inevitably specialized to specific applications. On the other hand, the non-parametric system is less structured and more applicable to a wider range of problems.

The work directly related to the research presented in this paper is in the area of the non-parametric design of array processors. Within this classification there are a number of approaches to the design problem. The most promising statistical procedures are those which can keep pace with the incoming data so as to constantly update the receiver's current state of knowledge about its environment [17] - [52]. There are primarily two approaches in changing the processor's parameters in "real-time": stochastic approximation [17] - [40] and adaptive estimation [41] - [52].

The application of these two theories to processor design in minimum-mean-squared error estimation problems yields a differential correction algorithm based on the method of steepest descent [8] - [9]. The form of the algorithm is

$$W_{n+1} = W_n - \mu_n Z_n \,, \tag{2.1}$$

where W_n is an estimate of W_n^*, μ_n is the convergence factor for the n^{th} iteration, and Z_n is an estimate of $J_n(W_n)$, the gradient of the mean-squared error surface $\xi_n(W)$ with respect to W evaluated at $W = W_n$. (Methods for obtaining the estimate Z_n will be discussed later.)

The stochastic approximation version of the algorithm (2.1) is characterized by the sequence $\{\mu_n\}$ tending to zero in some prescribed manner; the adaptive estimation version of the algorithm (2.1) is characterized by the sequence $\{\mu_n\}$ being set equal to some prescribed positive constant μ. The former procedure is designed to estimate the unknown parameters W in a strong probabilistic sense (mean-square and almost surely), while the latter is designed to allow a "tolerance" in the estimates. As will be shown in this research, by allowing convergence in a weak sense, the class of nonstationary problems that can be handled by adaptive estimation theory is larger than those handled by stochastic approximation theory.

The history of stochastic approximation theory began with its introduction in 1951 by Robbins and Monro [17]. The results directly related to the present research were obtained by Blum, Gardner, Dupac, and DeFigueridо. In 1954, Blum [18] extended the Robbins and Monro procedure to the estimation of a multivariate parameter. This permitted the application of stochastic approximation theory to the analysis of the gradient descent algorithms in optimization theory [19] - [40]. Gardner [21] demonstrated the applicability of this approach in the design of adaptive predictors.

The development of stochastic approximation algorithms for estimating a time-varying parameter received little attention until 1965 when Dupac [22] published his classic paper on dynamic stochastic approximation methods. DeFigueridо [24] extended this work to the estimation of a multivariable parameter evolving in a nonlinear fashion.

The development of the LMS adaptation algorithm was motivated by Widrow in considering determinstic gradient procedures for use in pattern recognition [41]. The LMS algorithm was later applied to adaptive filtering by Widrow [42] and by Widrow et al. [43], because, in part, of its conjectured ability to track nonstationarities. Griffiths [44] later modified the algorithm for certain array applications. Senne [46] provided an exact analysis of both Widrow's and Griffiths' algorithms under a special set of

assumptions (stationary, jointly Gaussian statistics). However, the technique used in Senne's analysis has not been shown to generalize for non-Gaussian, time-varying statistics. Daniell [50] - [51] has demonstrated an approach to the problem that can be generalized. It is this approach that will be extended in this research.

III. THE ADAPTATION ALGORITHM

A. THE DERIVATION OF THE ALGORITHM

The starting point for the derivation of the algorithm to be considered in this research is the procedure given in Chapter II by

$$W_{n+1} = W_n - \mu_n Z_n \ , \tag{3.1}$$

where W_n is an estimate of W_n^*, μ_n is the convergence factor for the n^{th} iteration, and Z_n is an estimate of $J_n(W_n)$, the gradient of the mean-squared error surface $\xi_n(W)$ with respect to W evaluated at $W = W_n$. (A typical choice for Z_n is [ ]

$$Z_n = -2(d_n - W_n^T X_n)X_n \ . \tag{3.2}$$

However, a number of other choices have been considered in the literature [42] - [50]. For a further discussion on methods of obtaining Z_n, the reader is referred to Appendix E, Section B.)

The important thing to note about Z_n, for the moment, is that if it were a good estimate of $J_n(W_n)$, then one would expect W_{n+1}, given by (2.1), to be a better estimate of W_n^* than W_n. Recall that a function changes most rapidly in the direction given by its gradient. Hence, by moving along the gradient, one is moving down the quadratic surface. However, in the nonstationary case, one would rather have W_{n+1} as a good estimate of

W^*_{n+1} , since the next data vector processed by the receiver is X_{n+1} . W^*_{n+1} is the optimum weight vector for this input. Motivated by the Kalman-Bucy theory [56], the following argument is presented for modifying (2.1) in order to track the sequence $\{W^*_n\}$.

Let us assume that the input nonstationarity is such that the optimum weight-vector sequence $\{W^*_n\}$ can be modeled by the discrete-time system

$$W^*_{n+1} = F_n(W^*_n) + U_n \ , \tag{3.3}$$

where $F_n(\cdot)$ is some function, not necessarily known, and $\{U_n\}$ is a zero-mean random process with finite second moment given by

$$E[\|U_n\|^2] = \rho_n^2 < \infty \ .^{\dagger}$$

According to this model, which is similar in form to the usual linear discrete-time model introduced by Kalman [55], the optimum weight vector undergoes a first-order Markov random walk. The problem for the adaptive process is to track W^*_n . The problem here differs from the Kalman problem in that here $F_n(\cdot)$ need not be linear and the random process $\{U_n\}$ need not be an independent Gaussian random process. (More will be said about this model in the next section.)

Since $\{U_n\}$ is zero-mean, from (3.3) one sees that $F_n(W^*_n)$ is an unbiased estimate of W^*_{n+1}. Hence, if a weight vector W were a good estimate of W^*_n , the one would expect

---

$^{\dagger}\|\cdot\|$ is the Euclidean norm defined by $\|U\| = U^TU$.

that $F_n(W)$ would be a good estimate of W^*_{n+1}. Therefore, a logical modification of the algorithm (2.1) is

$$W_{n+1} = F_n(W_n - \mu_n Z_n) \quad . \tag{3.4}$$

Unfortunately, in many applications, one will not have a priori knowledge of the sequence of functions $\{F_n\}$ or the nonstationarity cannot be modeled by (3.3). It may still be desirable to use an algorithm of the form (3.4). Let $\{G_n\}$ be a sequence of functions determined by the experimentor. (G_n could be an estimate of F_n, based on physical measurements, for example, or on a priori knowledge.) The algorithm (3.4) becomes

$$W_{n+1} = G_n(W_n - \mu_n Z_n) \quad . \tag{3.5}$$

(Some specific algorithms of the form (3.5) are given in Appendix E.) This is the adaptation algorithm to be considered in this research.

The procedure (3.5) is similar to the algorithm proposed by DeFigueiredo [24] for learning the unknown mixture distribution in a pattern recognition problem in which the environment is allowed to evolve in time. However, his formulation requires exact knowledge of the nonstationarity, i.e., both $\{F_n\}$ and $\{U_n\}$ are assumed known. Chein and Fu [23] also considers a similar problem to that of DeFigueiredo. Here again the nonstationarity must be known exactly.

The analysis of (3.5) given in Chapter IV of this research requires exact knowledge of only $\{F_n\}$ and $\{\rho_n^2\}$. The worst-case analysis for (3.5) presented in Chapter V removes even this restriction.

B. PRELIMINARIES

1. Assumptions

All vectors in this paper are contained in the Euclidean p-space E^p .

It is easily verified that the gradient $J_n(W) = R_n(W - W_n^*)$ of the mean-squared-error surface $\xi_n(W)$ satisfies the two conditions:

Condition (C1). For all $W \in R^p$

$$\|J_n(W)\|^2 \leq \lambda_{max}^2(n)\|W - W_n^*\|^2 , \tag{3.6}$$

and:

Condition (C2). For all $W \in R^p$

$$(W - W_n^*)^T J_n(W) \geq \lambda_{min}(n)\|W - W_n^*\|^2 , \tag{3.7}$$

where $\lambda_{min}(n)$ and $\lambda_{max}(n)$ are the minimum and maximum eigenvalues of R_n respectively.

It will be assumed that there exist positive constants λ_* and λ^* such that for all n:

Condition (C3).

$$0 < \lambda_* \leq \lambda_{min}(n) \leq \lambda_{max}(n) \leq \lambda^* < \infty . \tag{3.8}$$

In one-dimension these conditions require that $J_n(W)$ be bounded between the lines $\lambda_*(W - W_n^*)$ and $\lambda^*(W - W_n^*)$. (See Fig. 3.1.)

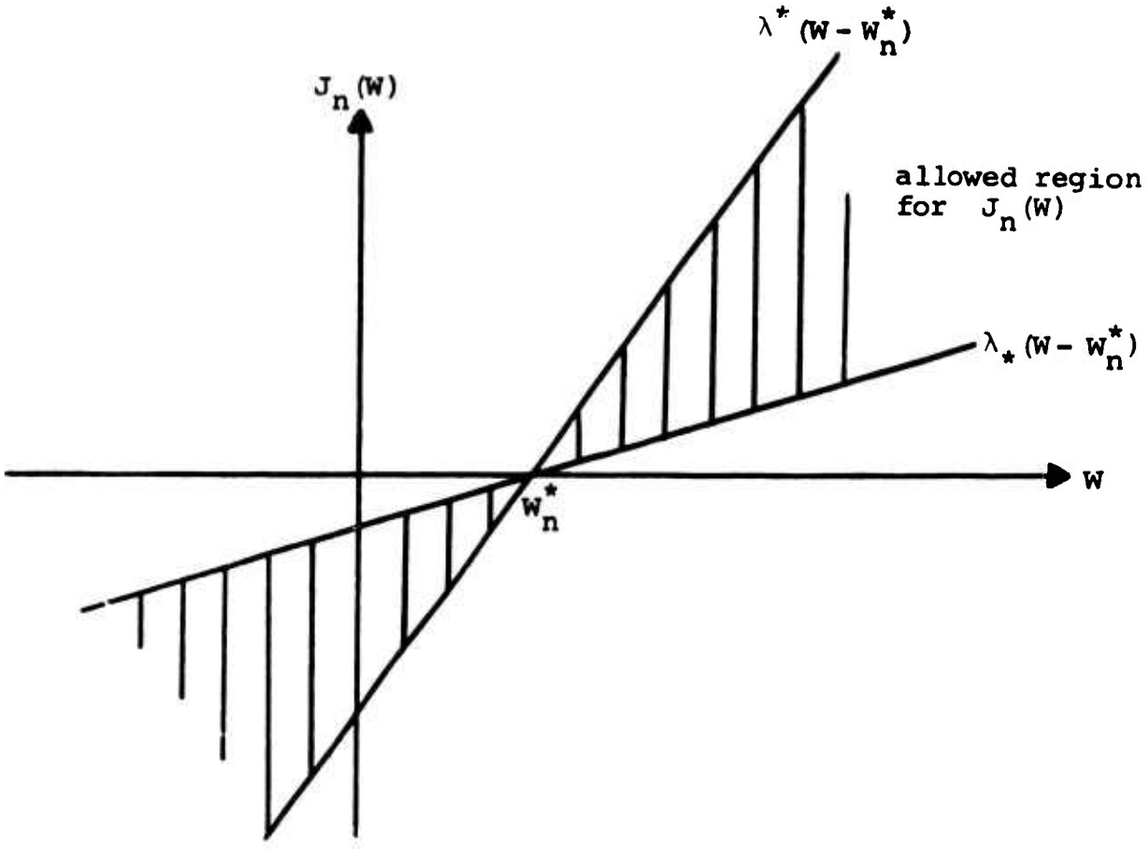


Fig. 3.1. Illustration of Conditions Placed on $J_n(W)$.

We assume that the gradient is measured in such a way that the sequence of gradient estimates $\{Z_n\}$ satisfies the two conditions:

Condition (C4). For all n

$$E[Z_n | W_n, W_n^*] = J_n(W_n) \; ; \tag{3.9}$$

Condition (C5). There exist positive constants σ_1 and σ_2 such that for all n

$$E[\|Z_n - J_n(W_n)\|^2 | W_n, W_n^*] \leq \sigma_1^2 + \sigma_2^2 \|W_n - W_n^*\|^2 \; . \tag{3.10}$$

Condition (C4) is the requirement that the gradient measurement be conditionally unbiased. This is consistent with the desire that algorithm (2.1) be based on the method of steepest descent. Condition (C5) reflects the experience that as one is farther from the optimum, the instantaneous mean-squared error becomes a noisier estimate of the expected mean-squared error. (For a zero-mean Gaussian random variable, the variance in its second moment is twice the second moment squared [42].) Hence, the instantaneous gradient estimate should be expected to increase.

Condition (C5) is satisfied easily by most gradient estimates (see Appendix E, Section C). The condition (C4) on the other hand is almost never satisfied exactly. In order for (C4) to hold, one essentially has to require that

the data pair (d_n, X_n) be conditionally independent of the previous data pairs. This insures that (d_n, X_n) is independent of W_n. However, it turns out that for a large weight system $(p \gg 1)$, condition (C4) is a reasonable assumption.

The model used in Chapter IV for the nonstationarity is that given by (3.3),

$$W_{n+1}^* = F_n(W_n^*) + U_n \ , \tag{3.3}$$

where $\{U_n\}$ is a zero-mean random process, not necessarily an independent Gaussian random process. The evolution transformation F_n also need not be linear. However, it will be assumed that F_n satisfies the Lipschitz condition

$$\sup \frac{\|F_n(W) - F_n(V)\|^2}{\|W - V\|^2} = f_n^2 < \infty \ , \tag{3.11}$$

where the supremum is over all weight vectors W and V. This condition is weaker than one requiring that the derivative of $F_n(W)$ exist and be bounded for all W. Note that the condition (3.11) does require that F_n be continuous in W.

The purpose of the Lipschitz condition is to bound the the maximum change or stretching of E^p allowed under F_n. If two vectors W and V are close together, one wants the transformed vectors $F_n(W)$ and $F_n(V)$ to be close together. Another way of putting it is if two vectors are close together, the effect of F_n operating on them should be similar.

It will also be assumed that the functions $G_n(\cdot)$ satisfy the Lipschitz condition,

$$\sup \frac{\|G_n(W) - G_n(V)\|^2}{\|W - V\|^2} \triangleq g_n^2 < \infty \ , \tag{3.12}$$

where the supremum is over all weight vectors W and V.

2. Mathematical Approach to the Analysis of (3.5)

The straightforward approach to the analysis of the algorithm (3.5) would be to develop a recursive relation for $E[\xi_n(W_n)]$, the expected mean-squared error, and evaluate this expression. However, this approach suffers the drawback that it leads to the setting up of the problem in a randomly time-varying metric space. It is worthwhile to pause a moment and see how this comes about.

Starting from (1.3) and using (1.4), it can be shown [ ] after some (easy) algebra that

$$\xi_n(W_n) = \xi_n^* + (W_n - W_n^*)^T R_n (W_n - W_n^*) \ . \tag{3.13}$$

Note that

$$\xi_n(W_n) - \xi_n^* = (W_n - W_n^*)^T R_n (W_n - W_n^*)$$

is the excess mean-squared error due to using the weight vector W_n rather than the optimum one W_n^*. The expected value of this expression, defined by

$$c_n^2 \triangleq E[(W_n - W_n^*)^T R_n (W_n - W_n^*)] \ , \tag{3.14}$$

is the expected excess mean-squared error. It is a measure of the cost associated with not having the necessary a priori knowledge to compute W_n^*. This quantity C_n^2 provides a measure of the efficiency of the adaptive algorithm (3.5). A large excess cost C_n^2 would indicate that the algorithm is not tracking the sequence $\{W_n^*\}$ very well, while a small excess cost C_n^2 would indicate that the algorithm is working well. A recursive relation for C_n^2 is desirable.

Unfortunately, the expression

$$(W_n - W_n^*)^T R_n (W_n - W_n^*)$$

is the equivalent to defining a random time-varying norm on the weight-space because R_n varies. This adds a further complication to the problem because the changes in both R_n and W_n^* have to be characterized in order to develop a recursive relation for (3.14). Note, however, that using assumption (3.8), one can obtain the inequalities†

$$\lambda_* E[\|W_n - W_n^*\|^2] \leq E[(W_n - W_n^*)^T R_n (W_n - W_n^*)] \leq \lambda^* E[\|W_n - W_n^*\|^2] \quad , \quad (3.15)$$

which follow from the trace inequalities [47] given by

† The inequalities (3.15) follow from the trace inequalities by noting

$$(W_n - W_n^*)^T R_n (W_n - W_n^*) = \mathrm{tr}\{R_n (W_n - W_n^*)(W_n - W_n^*)^T\} .$$

Make the identification $A = R_n$ and $B = (W_n - W_n^*)(W_n - W_n^*)^T$.

$$\lambda_{min}(A)tr\{B\} \leq tr\{AB\} \leq \lambda_{max}(A)tr\{B\} ,$$

where A and B are two symmetric, positive-semidefinite matrices of the same order and $\lambda_{min}(A)$ and $\lambda_{max}(A)$ are the minimum and maximum eigenvalues of A, respectively. Expressing bounds on the excess mean-squared error in terms of

$$b_n^2 \triangleq E[\|W_n - W_n^*\|^2] \tag{3.16}$$

avoids the randomly varying metric problem. For this reason, the expression (3.16) will be considered in this research.

Referring back to the algorithm (3.5), one sees that the sequence of random weight vectors $\{W_n\}$ depends on the choice of the sequence $\{\mu_n\}$. This sequence $\{\mu_n\}$ controls the stability and rate of convergence of the algorithm (3.5). In the following chapters the asymptotic properties of the sequence $\{b_n^2\}$ are investigated as a function of the choice of $\{\mu_n\}$.

IV. CONVERGENCE PROPERTIES OF THE ADAPTATION ALGORITHM

In this chapter, asymptotic properties of the sequence $\{b_n^2 \triangleq E[\|W_n - W_n^*\|]^2\}$ are investigated as a function of the convergence factors $\{\mu_n\}$ for the case in which the environmental functions $\{F_n\}$ are known. The corresponding adaptation algorithm to be used is (3.5) with $G_n = F_n$, given by

$$W_{n+1} = F_n(W_n - \mu_n Y_n) \ . \tag{3.5}$$

The first three results to be established below demonstrate the tracking ability of the constant-μ algorithm, while the last result provides sufficient conditions for the application of the corresponding stochastic approximation algorithm.

A. CONVERGENCE PROPERTIES OF THE CONSTANT-μ ALGORITHM

The following theorem and its proof provide many key results used in obtaining bounds on the sequence $\{b_n^2 \triangleq E[\|W_n - W_n^*\|^2]\}$ in the subsequent discussion. The theorem is:

Theorem 4.1. Assume that the optimum weight sequence $\{W_n^*\}$ is generated according to (3.3),

$$W_{n+1}^* = F_n(W_n^*) + U_n \ . \tag{3.3}$$

Assume

$$E[U_n] = 0$$

and

$$\rho^2 \triangleq \limsup_{n\to\infty} E[\|U_n\|^2] < \infty \ .$$

Let the adaptive processor be described by (3.5) with $G_n = F_n$,

$$W_{n+1} = F_n(W_n - \mu Y_n) \ . \tag{3.5}$$

Assume that the sequence of functions $\{F_n\}$ satisfy the Lipschitz condition (3.12) with

$$\limsup_{n\to\infty} f_n \triangleq f \leq 1 \ .$$

Assume that the sequence of gradient estimates $\{Z_n\}$ satisfy conditions (3.9) and (3.10), which are

$$E[Z_n | W_n, W_n^*] = J_n(W_n) \tag{3.9}$$

and

$$E[\|Z_n - J_n(W_n)\|^2 | W_n, W_n^*] \leq \sigma_1^2 + \sigma_2^2 \|W_n - W_n^*\|^2 \ . \tag{3.10}$$

Define

$$b_n^2 \triangleq E[\|W_n - W_n^*\|^2] \ . \tag{3.16}$$

Then, if

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \ ,$$

one can conclude

$$\limsup_{n\to\infty} b_n \leq \frac{\mu\sigma_1 f + \rho}{1 - f[1 - 2\mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2)]^{\frac{1}{2}}} \ . \tag{4.1}$$

Remark: Note that this bound applies even for the case in which the stochastic driving sequence $\{U_n\}$ is correlated in time. Moreover, the sequence $\{U_n\}$ may be correlated with either the optimum weight sequence $\{W_n^*\}$ or the adaptive weight sequence $\{W_n\}$ or both. The only requirement is that $\{U_n\}$ be zero-mean and asymptotically bounded in expected norm-squared. An example of this type of environment is one that can be modeled as a finite-state Markov dependent switching environment.

The general form of the bound (4.1) is shown in Fig . 4.1 for the two cases $f = 1$ and $f < 1$, respectively. Note that in both cases, the convergence factor μ which minimizes the lim sup of $\{b_n\}$ is different from zero. This is due to the unknown component, $\{U_n\}$, of the nonstationarity. More will be said about this effect later in discussing Corollary 4.1.1 and 4.1.2.

However, despite the general applicability of the bound, the real importance of this theorem lies in its proof. The methodology used here demonstrates the power of the formulation developed in Chapter III.

Proof of Theorem 4.1.

Subtract W_{n+1}^*, as given by (3.3) from both sides of (3.5) to obtain

$$W_{n+1} - W_{n+1}^* = F_n(W_n - \mu Z_n) - F_n(W_n^*) - U_n \ . \qquad (4.2)$$

Using Minkowski's inequality [61], which states that

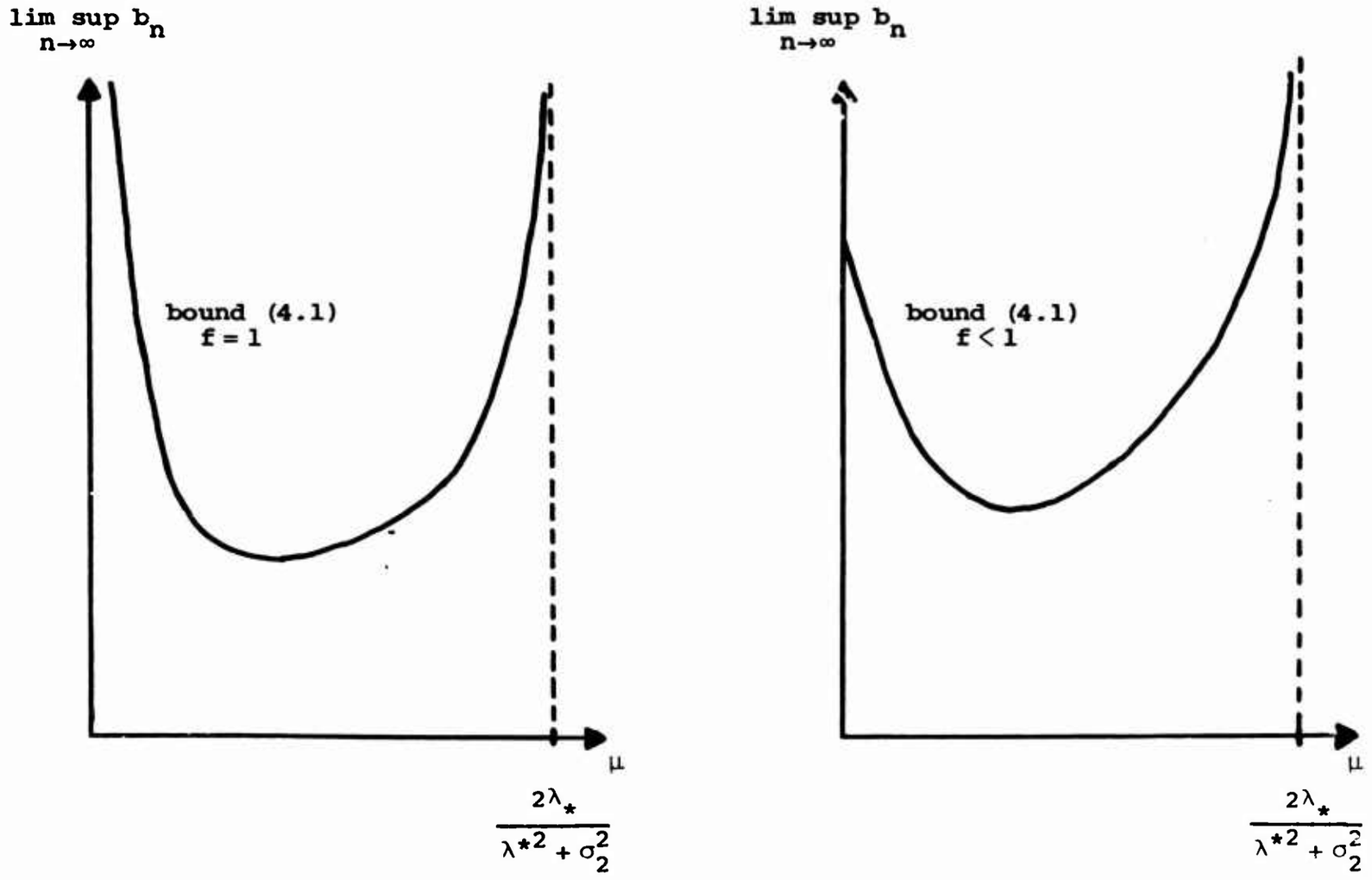


Fig. 4.1. Representative Curves for Bound (4.1)

$$\{E[(X+Y)^2]\}^{\frac{1}{2}} \leq \{E[X^2]\}^{\frac{1}{2}} + \{E[Y^2]\}^{\frac{1}{2}} , \qquad (4.3)$$

one concludes that based on (4.2)

$$b_{n+1} \leq \{E[\|F_n(W_n - \mu Z_n) - F_n(W_n^*)\|^2]\}^{\frac{1}{2}} + \rho_n , \qquad (4.4)$$

where

$$b_n^2 = E[\|W_n - W_n^*\|^2] ,$$

and

$$\rho_n^2 = E[\|U_n\|^2] .$$

The evaluation of (4.4) proceeds as follows: By the Lipschitz condition on F_n, given by (3.12), it follows that

$$\|F_n(W_n - \mu Z_n) - F_n(W_n^*)\|^2 \leq f_n^2 \|W_n - W_n^* - \mu Z_n\|^2$$

$$\leq f_n^2 \left\{ \|W_n - W_n^*\|^2 - 2\mu (W_n - W_n^*)^T Y_n + \mu^2 \|Z_n\| \right\} . \qquad (4.5)$$

By (3.9) and (3.7) it follows that

$$E\left[(W_n - W_n^*)^T Z_n\right] = E\left[(W_n - W_n^*)^T E[Z_n | W_n, W_n^*]\right]$$

$$= E[(W_n - W_n^*)^T J_n(W_n)]$$

$$\geq \lambda_* E[\|W_n - W_n^*\|^2] .$$

By (3.9), (3.10) and (3.6) it follows that

$$E\left[\|Z_n\|^2\right] = E\left[\|Z_n - J_n(W_n) + J_n(W_n)\|^2\right]$$

$$= E\left[\|Z_n - J_n(W_n)\|^2\right] + E\left[\|J_n(W_n)\|^2\right]$$

$$+ 2E\left[J_n^T(W_n) E[Z_n - J_n(W_n) | W_n, W_n^*]\right]$$

$$\leq \sigma_1^2 + (\sigma_2^2 + \lambda^{*2}) E\left[\|W_n - W_n^*\|^2\right] . \qquad (4.7)$$

Putting Eqs. (4.5), (4.6), and (4.7), together, one has the result

$$E\left[\|F_n(W_n - \mu Z_n) - F_n(W_n^*)\|^2\right] \leq f_n^2\left[1 - 2\mu\lambda_* + \mu^2(\sigma_2^2 + \lambda^{*2})\right] b_n^2 + \mu^2 f_n^2 \sigma_1^2 \ . \tag{4.8}$$

Once more using the Minkowski inequality, Eq. (4.3), conclude that:

$$\left\{E\left[\|F_n(W_n - \mu Z_n) - F_n(W_n^*)\|^2\right]\right\}^{\frac{1}{2}} \leq f_n\left[1 - 2\mu\lambda_* + \mu^2\left(\sigma_2^2 + \lambda^{*2}\right)\right]^{\frac{1}{2}} b_n + \mu f_n \sigma_1 \ .$$

From the above and (4.4), the key recursive relation,

$$b_{n+1} \leq f_n\left[1 - 2\mu\lambda_* + \mu^2\left(\sigma_2^2 + \lambda^{*2}\right)\right]^{\frac{1}{2}} b_n + \mu f_n \sigma_1 + \rho_n \ , \tag{4.9}$$

follows. Iterating (4.9) backward to $k = N$ yields

$$b_{n+1} \leq \left[\prod_{k=N}^{n} \alpha_k\right] b_N + \sum_{k=N}^{n}\left[\prod_{j=k+1}^{n} \alpha_j\right] \beta_k \ , \tag{4.10}$$

where

$$\alpha_k = f_k\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]^{\frac{1}{2}} \ ,$$

$$\beta_k = \mu f_k \sigma_1 + \rho_k \ ,$$

and

$$\prod_{k=n+1}^{n} \alpha_k = 1 \ .$$

By definition of limit superior [58], one has the result that for any $\varepsilon > 0$, there exists an N_ε such that for all $n \geq N_\varepsilon$

$$f_n < f + \varepsilon \ ,$$

and

$$\rho_n < \rho + \varepsilon \ .$$

Pick the ε such that for any

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2}$$

it is also the case that

$$\max\left[0, \frac{2}{\lambda_*}\left(1 - \frac{1}{f+\varepsilon}\right)\right] < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} - \max\left[0, \frac{2}{\lambda_*}\left(1 - \frac{1}{f+\varepsilon}\right)\right] \ ,$$

where

$$\max[x,y] \triangleq \begin{cases} x & x \geq y \\ y & y \geq x \end{cases} \ .$$

This guarantees that

$$(f+\varepsilon)\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]^{\frac{1}{2}} < 1 \ .$$

Hence, for any $n \geq N_\varepsilon$, one has from (4.10)

$$b_{n+1} < \alpha^{n+1-N_\varepsilon} b_{N_\varepsilon} + \sum_{k=N_\varepsilon}^{n} \alpha^{n-k} \beta$$

$$= \alpha^{n+1-N_\varepsilon}\left[b_{N_\varepsilon} - \frac{\beta}{1-\alpha}\right] + \frac{\beta}{1-\alpha} \ ,$$

where

$$\alpha = (f+\varepsilon)\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]^{\frac{1}{2}}$$

and

$$\beta = (f+\varepsilon)\mu\sigma_1 + \rho + \varepsilon \ .$$

Since $\alpha < 1$, for all $\delta > 0$, there exists an $M_\delta > N_\varepsilon$ such that for all $n \geq M_\delta$

$$\left| \alpha^{n-N_\varepsilon}\left[b_{N_\varepsilon} - \frac{\beta}{1-\alpha}\right] \right| < \delta \ .$$

Therefore, for all $n \geq M_\delta$,

$$b_{n+1} < \frac{\beta}{1-\alpha} + \delta < \frac{\mu\sigma_1 f + \rho}{1 - f\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]^{\frac{1}{2}}} + \gamma$$

where $\gamma > 0$ can be made arbitrarily small by choosing a sufficiently small ε and δ . From this it follows by definition of limit superior

$$\limsup_{n\to\infty} b_n \leq \frac{\mu\sigma_1 f + \rho}{1 - f\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]^{\frac{1}{2}}} \qquad (4.1)$$

<u>This completes the proof of Theorem 4.1.</u>

Two important special cases of the problem handled by the previous theorem are when:

(i) the nature of the nonstationarity becomes determinsistic in time (the random driving process $\{U_n\}$ goes to zero in expected norm-squared; i.e., $\rho = 0$). Examples of this case are:

a) stationary statistics. Here $F_n(W) = W$ and $U_n = 0$ for all time. This is the customarily treated problem in adaptive system theory [42]-[47].

b) asymptotically stationary statistics. Noise sources may be initially present that eventually move out of range of the receiver.

c) known varying channel. Measurements can be performed on the channel so as to determine its effect on the input statistics to the receiver.

d) known constraints placed on the weight vector. It may be desirable, for example, to control the frequency response of the array processor in a given direction while nulling out signals coming from other directions (see [40] or [48] or Appendix E, Section B).

(ii) the nature of the nonstationarity is strictly first-order Markoff (the random driving process $\{U_n\}$ is a zero-mean, independent random process). An example of this problem is where the input data x is the output of a linear randomly-time-varying channel with additive white noise [53] - [54]. The object of the filter is to predict the next input x_{n+1} on the bases of the previous p inputs.

The theorems are:

Corollary 4.1.1. Under the hypothesis of Theorem 4.1, if $\rho = 0$, then

$$\limsup_{n\to\infty} b_n^2 \leq \frac{\mu^2\sigma_1^2 f^2}{1 - f^2\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]} \tag{4.11}$$

and

Corollary 4.1.2. Under the assumptions of Theorem 4.1, if $\{U_n\}$ in (3.3) is a zero-mean, independent random process, then

$$\limsup_{n\to\infty} b_n^2 \leq \frac{\mu^2\sigma_1^2 f^2 + \rho^2}{1 - f^2\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]} . \tag{4.16}$$

Remark 1: The bounds (4.11) and (4.16) are shown in Figs. 4.2 and 4.3, respectively. While the form of the bounds (4.1) and (4.16) are similar for a given set of parameters, the bound (4.16) is tighter than that of (4.1).

Remark 2: One should note that by choosing a sufficiently small μ, the bound (4.11) can be made arbitrarily close to zero, i.e.,

$$\lim_{\mu \to 0^+} \limsup_{n \to \infty} b_n^2 = 0 .$$

An algorithm with this property is said to be ϵ-optimal [51].

Remark 3: For the stationary statistics problem in which $F_n(W) = W$ and $U_n = 0$, the bound given by (4.11) reduces to

$$\limsup_{n \to \infty} b_n^2 \leq \frac{\mu \sigma_1^2}{2\lambda_* - \mu\left(\lambda^{*2} + \sigma_2^2\right)}$$

This is the result given by Daniell [50].

Proof of Corollary 4.1.1.

From the proof of Theorem 4.1 (see (4.4) and (4.8)), it follows that

$$b_{n+1} \leq f_n\left\{\left[1 - 2\mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2)\right] b_n^2 + \mu^2\sigma_1^2\right\}^{\frac{1}{2}} + \rho_n \cdot \quad (4.12)$$

By Theorem 4.1, $\limsup_{n \to \infty} b_n$ is finite for $0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2}$.

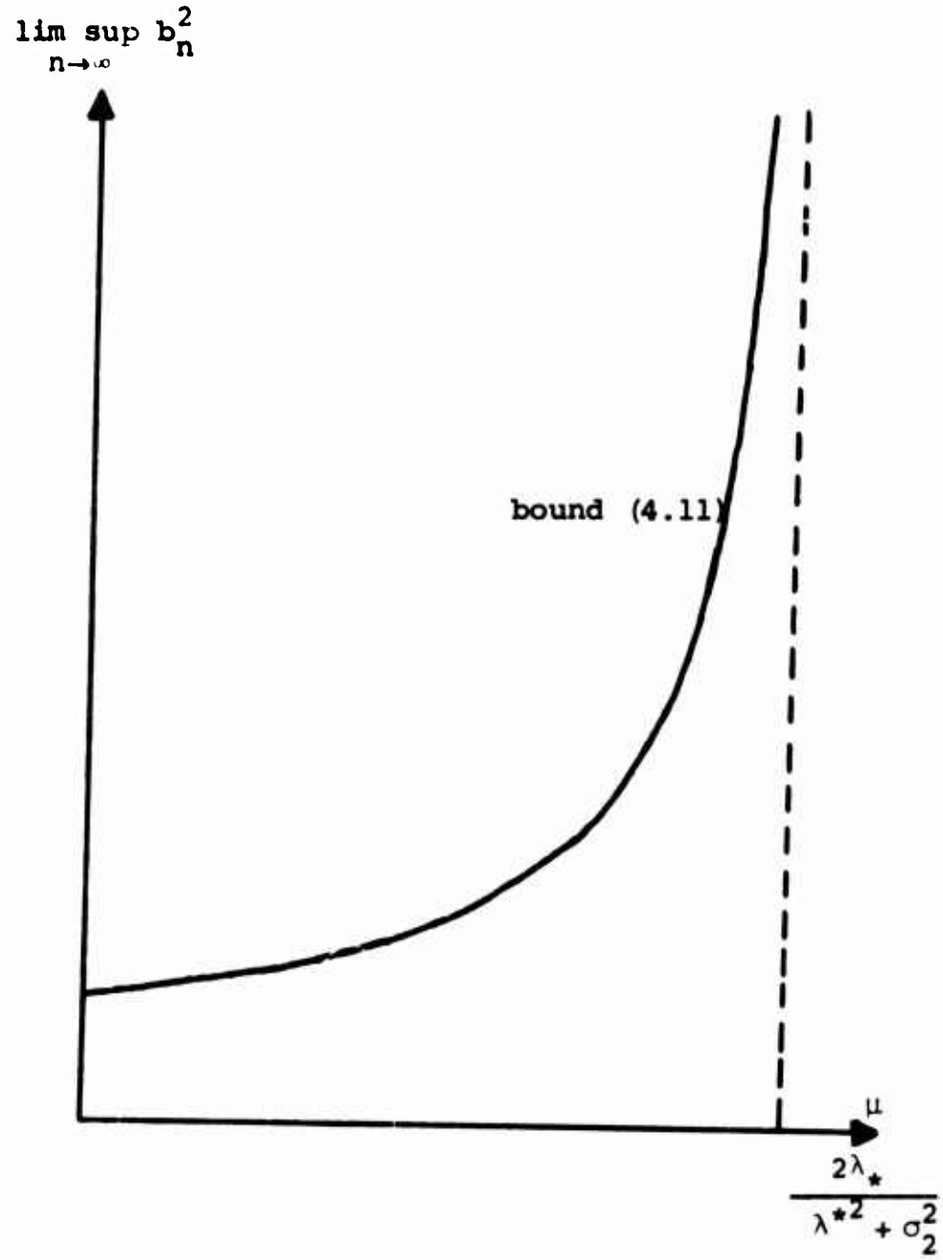


Fig. 4.2. Representative Curve for Bound (4.11).

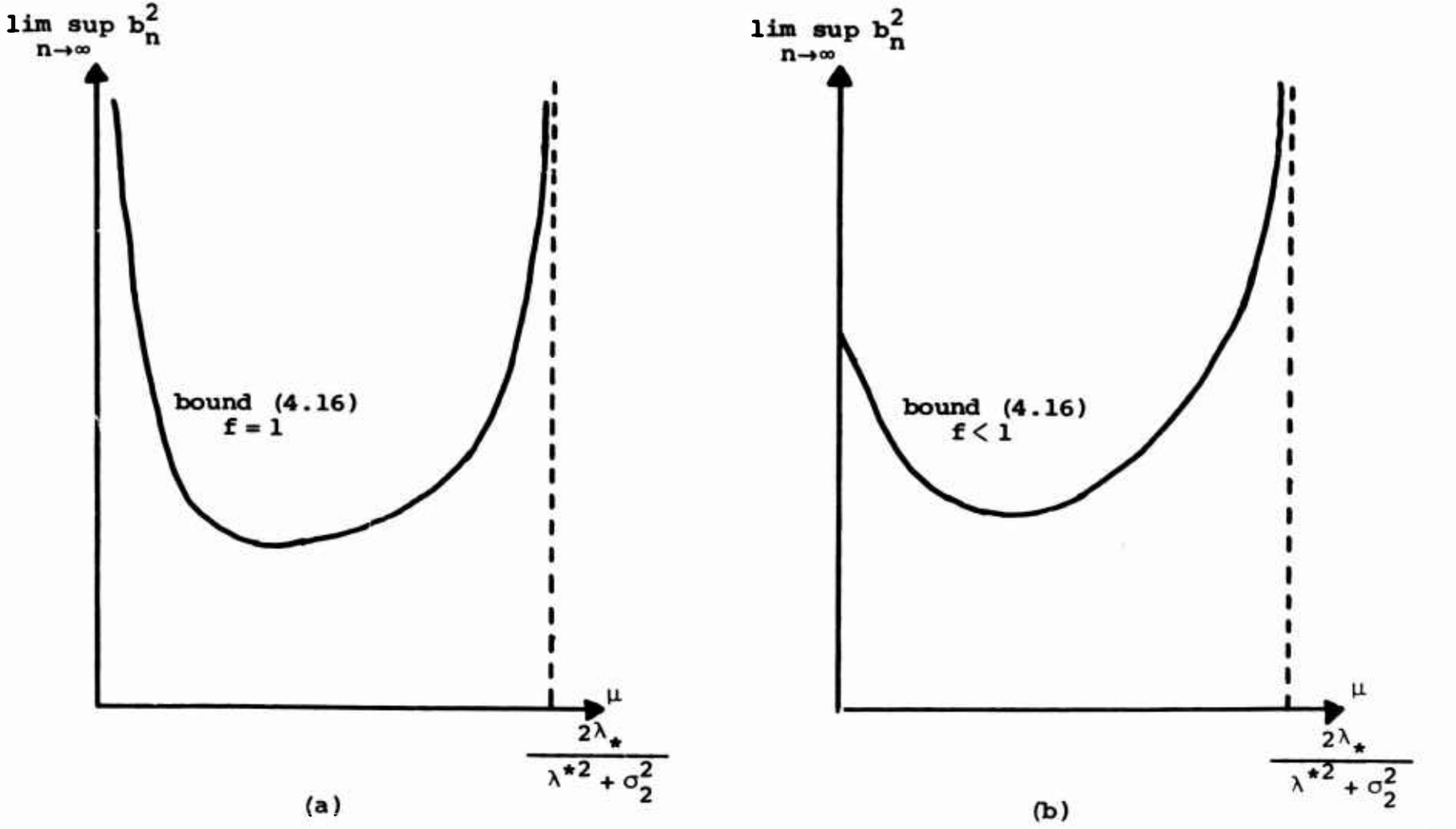


Fig. 4.3. Representative Curves for Bound (4.16)

Using the result, proven in Appendix A, that

$$\left(\limsup_{n\to\infty} b_n\right)^2 = \limsup_{n\to\infty} b_n^2$$

it then follows from (4.12) that for any $\varepsilon > 0$, there exists an N_ε such that for $n \geq N_\varepsilon$,

$$b_{n+1}^2 \leq f^2\left[1 - 2\mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2)\right]b_n^2 + \mu^2 f^2 \sigma_1^2 + \varepsilon \quad . \quad (4.13)$$

Noting the similarity of (4.9) and (4.13), one has by analogy to (4.1)

$$\limsup_{n\to\infty} b_n^2 \leq \frac{\mu^2 f^2 \sigma_1^2}{1 - f^2\left[1 - 2\mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2)\right]}$$

This completes the proof of Corollary 4.1.1.

Proof of Corollary 4.1.2.

Square both sides of (4.2) and take the expectation to obtain

$$E[\|W_{n+1} - W_{n+1}^*\|^2] = E[\|F_n(W_n - \mu Z_n) - F_n(W_n^*)\|^2] + E[\|U_n\|^2] - 2E[\{F_n(W_n - \mu Z_n)\}^T U_n] - 2E[\{F_n(W_n^*)\}^T U_n] \quad . \quad (4.17)$$

By the independence assumption on the sequence $\{U_n\}$, it follows that

$$E[\{F_n(W_n - \mu Z_n)\}^T U_n] = \{E[F_n(W_n - \mu Z_n)]\}^T E[U_n]$$

$$= 0 \ ,$$

and

$$E[\{F_n(W_n^*)\}^T U_n] = \{E[F_n(W_n^*)]\}^T E[U_n]$$
$$= 0 .$$

Recall (4.8) from the proof of Theorem 4.1,

$$E[\|F_n(W_n - \mu Y_n) - F_n(W_n^*)\|^2] \leq f_n^2\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]b_n^2 + \mu^2\sigma_1^2 f_n^2 \quad . \tag{4.8}$$

Therefore,

$$b_{n+1}^2 \leq f_n^2(1 - \mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2))b_n^2 + \mu^2\sigma_1^2 f_n^2 + \rho_n^2 \quad . \tag{4.18}$$

Comparing (4.18) with (4.9) in the proof of Theorem 4.1, one can argue by analogy that

$$\limsup_{n\to\infty} b_n^2 \leq \frac{\mu^2\sigma_1^2 f^2 + \rho^2}{1 - f^2\left[1 - 2\mu\lambda_* + \mu^2\left(\lambda^{*2} + \sigma_2^2\right)\right]} \tag{4.16}$$

<u>This completes the proof of Corollary 4.1.2.</u>

B. CONVERGENCE PROPERTIES OF THE STOCHASTIC APPROXIMATION ALGORITHM

For a subclass of the problems considered in the previous section, the sequence of random vectors $\{W_n\}$ generated by (3.5) converges in a strong probabilistic sense (e.g., mean-square convergence and probability one convergence) to the sequence $\{W_n^*\}$. The following theorem provides a set of sufficient conditions for convergence:

Theorem 4.2. Assume the optimum weights are generated by (3.3) where $\{U_n\}$ is a zero-mean independent random process with $E[\|U_n\|^2] = \rho_n^2$ where

$$\sum_{n=1}^{\infty} \rho_n^2 < \infty .$$

Let the adaptive processor be given by (3.5) with $G_n = F_n$. Assume $\{F_n\}$ satisfy (3.12). If $\{\mu_n\}$ is a sequence of non-negative real numbers satisfying

$$\sum_{k=1}^{\infty} \mu_k^2 < \infty$$

and if

(i) for sufficiently large n, $f_n^2(1 - \mu_n\lambda_*) \leq 1$

(ii) $$\sum_{k=1}^{\infty} (1 - f_k^2(1 - \mu_k\lambda_*)) = \infty ,$$

then

$$\lim_{n\to\infty} E[\|W_n - W_n^*\|^2] = 0$$

and

$$P[\lim_{n\to\infty} \|W_n - W_n^*\|^2 = 0] = 1 \ .$$

Proof of Theorem 4.2.

Since convergence in mean-square will be needed to show convergence with probability one, this aspect will be considered first.

Pick N_1 such that for all $n \geq N_1$,

$$0 \leq \mu_n \leq \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \ .$$

Then for $n \geq N_1$, one can obtain the equivalent expression to(4.18) in the proof of Corollary 4.1.2,

$$b_{n+1}^2 \leq \alpha_n b_n^2 + \beta_n \ , \tag{4.19}$$

where

$$\alpha_n = f_n^2\left[1 - 2\mu_n\lambda_* + \mu_n^2\left(\lambda^{*2} + \sigma_2^2\right)\right] \tag{4.20}$$

and

$$\beta_n = \mu_n^2 f_n^2 \sigma_1^2 + \rho_n^2 \ . \tag{4.21}$$

Therefore, by recursive substitution, one has for $n > N_1$

$$b_{n+1}^2 \leq \left[\prod_{K=N_1}^{n} \alpha_k\right] b_{N_1}^2 + \sum_{K=N_1}^{n}\left[\prod_{j=k+1}^{n} \alpha_j\right] \beta_k \ . \tag{4.22}$$

An especially straightforward proof of convergence of (4.22) is based on Kronecker's lemma, proven in Appendix B,

Lemma 4.2.1 (Kronecker's Lemma) Let $\{x_k\}$ be a sequence of real numbers. Let $\{a_k\}$ be a sequence of positive numbers converging monotonically upward to infinity. If $\sum_{K=1}^{n} \frac{x_k}{a_k} = s_n$ converges to some finite number, say s, then

$$\lim_{n\to\infty} \frac{1}{a_n} \sum_{K=1}^{n} x_k = 0 .$$

Returning to (4.22), assume, without a loss of generality, that $N_1 = 1$. Make the identification with Lemma 4.2.1, that

$$\frac{1}{a_k} = \prod_{j=1}^{k} \alpha_j$$

and

$$x_k = a_k \beta_k .$$

Therefore, if

$$\prod_{k=1}^{n} \alpha_k \searrow 0 \quad \text{as } n\to\infty \tag{4.23}$$

and if

$$\sum_{k=1}^{\infty} \left[\mu_k^2 f_k^2 \sigma_1^2 + \rho_k^2\right] < \infty , \tag{4.24}$$

then $\lim_{n\to\infty} b_n^2 = 0$.

To show (4.24), note that f_k^2 is bounded. Therefore, it follows that

$$\sum_{k=1}^{\infty} [\mu_k^2 f_k^2 \sigma_1^2 + \rho_k^2] < \infty$$

if

$$\sum_{k=1}^{\infty} \mu_k^2 < \infty \quad \text{and} \quad \sum_{k=1}^{\infty} \rho_k^2 < \infty \ .$$

To show (4.23), use the inequality, $e^{-x} \geq 1 - x$, for all x , to obtain

$$\prod_{k=1}^{n} \alpha_k \leq \exp\left\{ \sum_{k=1}^{n} (\alpha_k - 1) \right\} \ .$$

Therefore it suffices to show that there exists an N_2 such that for all $n \geq N_2$, $\alpha_n \leq 1$, and

$$\sum_{k=1}^{n} (1 - \alpha_k) \to \infty \quad \text{as} \quad n \to \infty$$

Now

$$1 - \alpha_k = 1 - f_k^2 \left[1 - 2\mu_k \lambda_* + \mu_k^2 \left(\lambda^{*2} + \sigma_2^2 \right) \right] \ .$$

Since $\mu_k \to 0$, one can assume without loss of generality that for all k

$$\alpha_k \leq f_k^2 (1 - \mu_k \lambda_*) \ .$$

By hypothesis of theorem, it immediately follows

$$\lim_{n \to \infty} E[\|W_n - W_n^*\|^2] = 0 \ .$$

To show convergence with probability one, take conditional expectations to obtain, in an analogous fashion to (4.19),

$$E[\|W_{n+1} - W_{n+1}^*\|^2] \, |W_n, W_n^*] \leq \alpha_n \|W_n - W_n^*\|^2 + \beta_n$$

where α_n and β_n are defined by (4.20) and (4.21). Therefore

$$E[\|W_{n+1}-W^*_{n+1}\|^2 \,|\, \|W_n-W^*_n\|^2] = E\left[E\|W_{n+1}-W^*_{n+1}\|^2 \,|W_n, W^*_n] \,|\, \|W_n-W^*_n\|^2\right]$$

$$\leq \alpha_n \|W_n-W^*_n\|^2 + \beta_n \,. \qquad (4.25)$$

Note that if $\alpha_n \leq 1$ and $\beta_n = 0$, then one could use the martingale convergence theorem to prove convergence of (4.25). However, $\beta_n \neq 0$. The following lemma, proven in Appendix C, provides the necessary result to show convergence.

Lemma 4.2.2. Let $\{X_k\}$ be a random process such that

(i) $\sup_k E[|X_k|] < \infty$

(ii) $E[X_{k+1}|X_k,\ldots,X_1] \geq X_k - a_k$ for all k.

where $\{a_k\}$ is a sequence of non-negative real numbers such that

$$\Sigma a_k < \infty \,.$$

Then with probability one,

$$\lim_{n\to\infty} X_n = X_\infty \quad \text{where} \quad E[|X_\infty|] < \infty \,.$$

Make the identification with Lemma 4.2.2,

$$-x_n = \|W_n - W^*_n\|^2$$

and

$$\beta_n = a_n \,.$$

By the first part of this proof, it has been shown that

$$\alpha_n \leq 1$$

$$\sum_{K=1}^{n} \beta_n < \infty$$

and

$$\lim_{n\to\infty} b_n^2 = 0 \ .$$

Hence, $\sup_n E\left[\|W_n - W_n^*\|^2\right] < \infty$. Therefore, by Lemma 4.2.2, one has with probability one

$$\lim_{n\to\infty} \|W_n - W_n^*\|^2 = x_\infty \ ,$$

where

$$E[\,|x_\infty|\,] < \infty \ .$$

To show $x_\infty = 0$, use Fatou's lemma [61], which states that

$$E\left[\lim_{n\to\infty} X_n\right] \leq \lim_{n\to\infty} E[X_n] \ ,$$

to obtain the chain of inequalities,

$$0 \leq E\left[\lim_{n\to\infty} \|W_n - W_n^*\|^2\right] \leq \lim_{n\to\infty} E\left[\|W_n - W_n^*\|^2\right] = 0 \ .$$

Thus, $\lim_{n\to\infty} \|W_n - W_n^*\|^2 = 0$ a.e.

This completes the proof of Theorem 4.2.

Remark: With $F_n(W) = W$ and $U_n = 0$ for all n , one has algorithm (3.5) in the stationary statistics case. By the previous theorem, if

$$\mu_n \geq 0$$

$$\Sigma \mu_n = \infty$$

$$\Sigma \mu_n^2 < \infty$$

then $W_n \rightarrow W^*$ in mean-square and with probability one (see Appendix E). These are the usual conditions required in the stochastic approximation literature [17] - [39]. However, one important difference here is the additional variance term, $\sigma_2^2 \|W_n - W_n^*\|^2$, given by (3.10). This term prevents the application of Dvoretzky's theorem [35] to prove convergence.

V. A WORST-CASE ANALYSIS

Implicit in the analysis presented in the previous chapter is that the nonstationarity is known and can be modeled by the discrete-time system (3.3). If the nonstationarity is unknown or cannot be modeled by (3.3), the results given in that chapter do not apply. In this chapter, bounds are obtained for the asymptotic behavior of the adaptive system (3.5) under mild restrictions on the optimum weight vector sequence $\{W_n^*\}$. It should be emphasized that the results given here are <u>not</u> limited to the nonstationarity model (3.3).

The three classes of nonstationarities considered are the bounded-increment, bounded-variation, and bounded-optimum. The bounded increment class is defined to consist of those sequences $\{W_n^*\}$ for which

$$\limsup_{n\to\infty} E[\|W_{n+1}^* - W_n^*\|^2] = \Delta^2 < \infty \quad , \tag{5.1}$$

the bounded variation class is defined to consist of those sequences $\{W_n^*\}$ for which

$$\lim_{n\to\infty} \sum_{k=1}^{n} \{E[\|W_{k+1}^* - W_k^*\|^2]\}^{\frac{1}{2}} < \infty \quad , \tag{5.2}$$

and the bounded optimum class is defined to consist of those sequences $\{W_n^*\}$ for which

$$\limsup_{n\to\infty} E[\|W_n^* - W_0^*\|^2] = B^2 < \infty \quad , \tag{5.3}$$

for some constant weight vector W_0^*.

The emphasis of the analysis is for the algorithm (3.5) with $G_n(W) = W$; i.e., the original algorithm (3.1). However, the theorems stated have analogies to the more general algorithm (3.5). (The extension to this case is straightforward.) The three major results given are by no means exhaustive of the possible situations that can be encountered. Nevertheless, they do illustrate some basic approaches and philosophy for the handling of the nonstationary statistics problem.

The analysis presented here is a worst-case analysis for (3.1) in the following sense. Pick the convergence factor sequence $\{\mu_n\}$ (μ_n could equal μ for all n). For each of the three previously mentioned nonstationarities the corresponding $\limsup_{n\to\infty} b_n^2$ is the asymptotic bound on the supremum of $E[\|W_n - W_n^*\|^2]$ over all possible input data pair sequences $\{(d_k, X_k)\}$ that satisfy the conditions (3.6) - (3.10) for a specific choice of λ_*, λ^*, σ_1, and σ_2. (In general, a different sequence $\{\mu_n\}$ will result in a different choice of the sequence $\{(d_k, X_k)\}$. However, the bound $\limsup_{n\to\infty} b_n^2$ may remain unchanged.) In other words, within a given set of constraints (conditions (3.6) - (3.10), $\{\mu_n\}$, and the class of nonstationarity), what is the most manevolent thing that nature can do to the behavior of the algorithm (3.1) in terms of the sequence $\{b_n^2\}$?

A. NONSTATIONARITY OF THE BOUNDED-INCREMENT CLASS

A surprising amount of information about the performance of the adaptive filter can be inferred from knowing only that the expected change in the norm of the optimum weight is bounded. This is sufficient to conclude that for suitable choices of the gain parameter μ, the optimum weight vector can be tracked within some finite distance. The result is summarized by

Theorem 5.1. For the adaptive filter system described in Chapter III with $G_n(W) = W$, if

$$\limsup_{n\to\infty} E[\|W^*_{n+1} - W^*_n\|^2] = \Delta^2 < \infty ,$$

and

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} ,$$

then

$$\limsup_{n\to\infty} b_n \leq \frac{\Delta + \mu\sigma_1}{1 - \sqrt{1 - 2\mu\lambda_* + \mu^2 (\lambda^{*2} + \sigma_2^2)}} \tag{5.4}$$

Proof of Theorem 5.1.

Starting from (3.5), with $G_n(W) = W$, subtract W^*_{n+1} from both sides to obtain

$$W_{n+1} - W^*_{n+1} = W_n - W^*_n - \mu Z_n - (W^*_{n+1} - W^*_n) .$$

Using the Minkowski inequality (4.3), one concludes

$$b_{n+1} \leq \sqrt{E[\|W_n - W^*_n - \mu Z_n\|^2]} + \Delta_n \tag{5.5}$$

where

$$b_n^2 = E[\|W_n - W_n^*\|^2]$$

$$\Delta_n^2 = E[\|W_{n+1}^* - W_n^*\|^2] \quad .$$

Comparing (5.5) to (4.4) in the proof of Theorem 4.1, one has the conclusion (5.4) by direct analogy to (4.1) with $f = 1$.

This completes the proof of Theorem 5.1.

It is interesting to note that by using the conclusion of this theorem, one can obtain a tighter bound by returning to its proof. This observation is summarized by

Corollary 5.1.1. Under the hypothesis of Theorem 5.1,

$$\limsup_{n\to\infty} b_n \leq \frac{\Delta}{1-\alpha^2} + \sqrt{\left(\frac{\Delta}{1-\alpha^2}\right)^2 + \frac{\beta^2 - 2\Delta^2}{1-\alpha^2}} \qquad (5.6)$$

where

$$\alpha^2 = 1 - 2\mu\lambda_* + \mu^2(\sigma_2^2 + \lambda^{*2})$$

and

$$\beta^2 = \Delta^2 + \mu^2\sigma_1^2 \quad .$$

Proof of Corollary 5.1.1.

Using (4.8) in (5.5), one obtains

$$b_{n+1} \leq [\alpha^2 b_n^2 + \mu^2\sigma_1^2]^{\frac{1}{2}} + \Delta_n \quad , \qquad (5.7)$$

where

$$\alpha^2 = 1 - 2\mu\lambda_* + \mu^2(\sigma_2^2 + \lambda^{*2}) \ .$$

Using the result proven in Appendix A, which states

$$\limsup_{n\to\infty} b_n^2 = \left(\limsup_{n\to\infty} b_n\right)^2 \ ,$$

and the conclusion of Theorem 5.1 that $\limsup_{n\to\infty} b_n$ is finite, one can conclude that for sufficiently large n

$$b_{n+1} \leq [\alpha^2 (b^*)^2 + \mu^2\sigma_1^2]^{\frac{1}{2}} + \Delta + \epsilon \ , \tag{5.8}$$

where

$$b^* \triangleq \limsup_{n\to\infty} b_n$$

and

$$\epsilon > 0 \quad \text{arbitrary} \ .$$

From (5.8) it follows directly that

$$b^* \leq [\alpha^2 (b^*)^2 + \mu^2\sigma_1^2]^{\frac{1}{2}} + \Delta \ . \tag{5.9}$$

Solving for b^* in (5.9), one obtains

$$b^* \leq \frac{\Delta}{1-\alpha^2} + \sqrt{\left(\frac{\Delta}{1-\alpha^2}\right)^2 + \frac{\mu^2\sigma_1^2 - \Delta^2}{1-\alpha^2}} \tag{5.6}$$

This completes the proof of Corollary 5.1.1.

Remark: It has been argued by Widrow [49] that the "rate of adaptation is optimized when the loss of performance resulting from adapting too rapidly equals twice the loss in performance resulting from adapting too slowly." Since the

rate of adaptation is inversely proportional to the gain-constant μ [42], an equivalent statement to the one above is, "the gain-constant μ is optimized when the loss of performance resulting from adapting too rapidly equals twice the loss in performance resulting from adapting too slowly." Under certain conditions, to be specified, this rule applies very closely to the bound (5.6), as shown by the following argument:

Assume that $b^*(\mu)$, given by (5.6), is minimized with respect to μ for

$$\mu << \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \quad .$$

For a value of μ satisfying this condition one can consider the bound

$$b^*(\mu) = \frac{\Delta}{2\mu\lambda_*} + \sqrt{\left(\frac{\Delta}{2\mu\lambda_*}\right)^2 + \frac{\mu\sigma_1^2}{2\lambda_*}} \qquad (5.10)$$

The component due to changes in the optimum weight vector is

$$b^*_{TV}(\mu) = \frac{\Delta}{\mu\lambda_*} \quad .$$

(Set $\sigma_1 = 0.$) The component due to noise in the gradient estimate is

$$b^*_{MN}(\mu) = \sqrt{\frac{\mu\sigma_1^2}{2\lambda_*}} \quad .$$

(Set $\Delta = 0.$)

Express $b^*(\mu)$ as

$$b^*(\mu) = \tfrac{1}{2} b^*_{TV}(\mu) + \sqrt{\left[\tfrac{1}{2} b^*_{TV}(\mu)\right]^2 + \left[b^*_{MN}(\mu)\right]^2} \quad . \qquad (5.11)$$

Taking the derivative of $b^*(\mu)$ with respect to μ and setting equal to zero, one obtains

$$2\left[b^*_{TV}(\mu_0)\right]^2 = \left[b^*_{MN}(\mu_0)\right]^2 \quad , \qquad (5.12)$$

where μ_0 is the value of the gain constant μ which minimizes $b^*(\mu)$ given by (5.11). Solving (5.12) for μ_0 yields

$$\mu_0 = \sqrt[3]{\frac{4\Delta^2}{\lambda_* \sigma_1^2}} \quad . \qquad (5.13)$$

Hence. if

$$\Delta^{2/3} << \frac{\lambda_*}{\lambda^{*2} + \sigma_2^2} \sqrt[3]{\lambda_* \sigma_1^2}$$

then the value of μ given by (5.12) and (5.13) is close to the value of μ which minimizes the bound (5.6). In other words, for a slowly varying environment, a good rule of thumb is pick the gain constant μ using Widrow's rule.

B. NONSTATIONARITY OF THE BOUNDED-VARIATION CLASS

It may readily be seen that Corollary 5.1.1 applies to the bounded variation problem since $\sum_k \Delta_k < \infty$ implies $\Delta_k \to 0$. However, within this class of nonstationarity is the stationary weight vector case, $W_k^* = W^*$ for all k, and the asymptotically stationary weight vector case, $W_k^* \to W^*$.† It will be shown that stochastic approximation algorithms ($\mu_n \to 0$) can also be applied with success, to these two cases. The result is summarized by:

Theorem 5.2. For the adaptive algorithm given in Chapter III with $G_n(W) = W$, if the sequence $\{\mu_n\}$ satisfy

$$\text{i)} \quad \mu_n \geq 0$$

$$\text{ii)} \quad \sum_n \mu_n = \infty$$

$$\text{iii)} \quad \sum_n \mu_n^2 < \infty$$

---

†The fact that $\sum_k \|W_{k+1}^* - W_k^*\| < \infty$ implies $\lim_{k\to\infty} W_k^* = W^*$ follows from the inequality ($m > n$),

$$\|W_n^* - W_m^*\| = \|\sum_{k=n}^{m-1} (W_{k+1}^* - W_k^*)\| \leq \sum_{k=n}^{m-1} \|W_{k+1}^* - W_k^*\|$$

and

$$\sum_n^\infty \|W_{k+1}^* - W_k^*\| \to 0 \quad \text{as} \quad n \to \infty \quad .$$

Hence, by the Cauchy convergence criterion [58], $W^* \triangleq \lim_{k\to\infty} W_k^*$ exists.

and the sequence of optimum weight vectors $\{W_n^*\}$ satisfy

$$\sum_n \Delta_n < \infty ,$$

where

$$\Delta_n = \|W_{n+1}^* - W_n^*\| ,$$

then

$$\lim_{n\to\infty} b_n^2 = 0$$

and

$$P\left\{\lim_{n\to\infty} \|W_n - W_n^*\| = 0\right\} = 1 .$$

Proof of Theorem 5.2.

Starting from $W_{n+1} = W_n - \mu_n Z_n$, subtract W_{n+1}^* from both sides, square, and take expectations to obtain

$$b_{n+1}^2 \le \alpha_n^2 b_n^2 + \mu_n^2 \sigma_1^2 + \Delta_n E[\|W_n - W_n^* - \mu_n Z_n\|] + \Delta_n^2$$

where

$$\alpha_n^2 = 1 - 2\mu_n \lambda_* + \mu_n^2 (\lambda^{*2} + \sigma_2^2)$$

and b_n^2 and Δ_n are as before. Use the inequality (Appendix E)

$$E[|x|] \le \varepsilon + \frac{1}{\varepsilon} E[x^2]$$

and (4.8) to conclude

$$b_{n+1}^2 \leq \left(1+\frac{\Delta_n}{\varepsilon}\right)\alpha_n^2 b_n^2 + \left(1+\frac{\Delta_n}{\varepsilon}\right)\mu_n^2\sigma_1^2 + \Delta_n^2 + \varepsilon\,\Delta_n \,.$$

Iterating backwards, one gets the result

$$b_{n+1}^2 \leq \left[\prod_{k=N}^{n}\left(1+\frac{\Delta_k}{\varepsilon}\right)\alpha_k^2\right] b_N^2 + \sum_{k=N}^{n}\left[\prod_{j=k+1}^{n}\left(1+\frac{\Delta_j}{\varepsilon}\right)\alpha_j^2\right]\beta_k^2 \,,$$

where

$$\beta_k^2 = \left(1+\frac{\Delta_k}{\varepsilon}\right)\mu_k^2\sigma_1^2 + \Delta_k^2 + \varepsilon\,\Delta_k \,.$$

As done in the proof of Theorem 4.2, one can use Kronecker's lemma (Lemma 4.2.1), to conclude that if

$$\sum_{k=1}^{\infty}\beta_k^2 < \infty$$

and for some N

$$\prod_{n\geq N}\left(1+\frac{\Delta_n}{\varepsilon}\right)\alpha_n^2 \searrow 0$$

then $\lim_{n\to\infty} b_n^2 = 0$. If

$$\sum_{k=1}^{\infty}\mu_k = \infty \,,$$

$$\sum_{k=1}^{\infty}\mu_k^2 < \infty \,,$$

and

$$\sum_{k=1}^{\infty}\Delta_k < \infty \,,$$

then one has $\sum_{k=1}^{\infty}\beta_k^2 < \infty$. Using the inequality $e^{-x} \geq 1-x$, it follows for sufficient large N ,

$$\prod_{k=N}^{n}\left(1+\frac{\Delta_k}{\varepsilon}\right)\alpha_k^2 \le \exp\left\{-\sum_{k=N}^{n} 1-\left(1+\frac{\Delta_k}{\varepsilon}\right)\alpha_k^2\right\}$$

$$\le \exp\left\{-\sum_{k=N}^{n} 1-\left(1+\frac{\Delta_k}{\varepsilon}\right)\left(1-\mu_k\lambda_*\right)\right\}$$

$$\le \exp\left\{-\sum_{k=N}^{n}\left[\mu_k\lambda_* - \frac{\Delta_k}{\varepsilon}\left(1-\mu_k\lambda_*\right)\right]\right\}$$

$$\le \exp\left\{-\frac{1}{2}\lambda_* \sum_{k=N}^{n}\mu_k\right\} \searrow_{n\to\infty} 0 .$$

Thus, $\lim_{n\to\infty} b_n^2 = 0$.

The probability one result follows in a manner similar to Theorem 4.2.

<u>This completes the proof of Theorem 5.2.</u>

C. NONSTATIONARITY OF THE BOUNDED OPTIMUM CLASS

It is often the case that the optimum weight vector is known to lie within a p-dimensional hypersphere. For example, if the background noise field fluctuates about some average value, the optimum will fluctuate about a fixed vector. The following theorem gives an upper bound for the sequence $\{b_n\}$ for this model.

Theorem 5.3. For the adaptive system described in Chapter II, if there exists a vector W_0^* and a positive constant B such that

$$\limsup_{n\to\infty} \|W_n^* - W_0^*\| \leq B ,$$

then

$$\limsup_{n\to\infty} b_n \leq 2B + \min_{\varepsilon\geq\varepsilon_0} \sqrt{\frac{\varepsilon + \mu\sigma_1^2}{(1-\frac{\varepsilon_0}{\varepsilon})\,[2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)]}} \qquad (5.14)$$

where

$$\varepsilon_0 = \frac{\lambda^{*2} B^2}{2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)} \qquad (5.15)$$

Proof of Theorem 5.3.

For the model defined by the hypothesis it is not possible in general to bound directly the sequence $\{b_n^2\}$. However, by considering the sequence $\{c_n^2\}$ defined by

$$c_n^2 = E[\,\|W_n - W_0^*\|^2\,]$$

one can infer an upper bound for $\{b_n^2\}$. The first step is to obtain a recursive relation for sequence $\{c_n^2\}$.

Subtracting W_0^* from both sides of (3.5), squaring, and taking expectations, yields

$$c_{n+1}^2 = c_n^2 - 2\mu E[\,(W_n - W_0^*)^T Z_n] + \mu^2 E[\,\|Z_n\|^2\,] .$$

The second term on the R.H.S. may be bounded as follows,

$$\begin{aligned} E[\,(W_n - W_0^*)^T Z_n] &= E[\,(W_n - W_n^*)^T E[Z_n | W_n, W_n^*]] \\ &= E[\,(W_n - W_0^*)^T J_n(W_n)] \\ &= E[\,(W_n - W_n^*)^T J_n(W_n)] + E[\,(W_n^* - W_0^*)^T J_n(W_n)] \\ &\geq \lambda_* b_n^2 - \tfrac{1}{2}(\varepsilon + \tfrac{1}{\varepsilon}\,\lambda^{*2} B^2 b_n^2) \quad . \end{aligned}$$

where by using the inequality (Appendix D),

$$E[\,|x|\,] \leq \frac{\varepsilon}{2} + \frac{1}{2\varepsilon}\,E[x^2] \quad , \quad \varepsilon > 0 \ \text{ arbitrary},$$

one has

$$E[\,(W_n^* - W_0^*)^T J_n(W_n)\,] \leq E[\,\|W_n^* - W_0^*\|\,\|J_n(W_n)\|\,] \leq \tfrac{1}{2}(\varepsilon + \tfrac{1}{\varepsilon}\lambda^{*2} B^2 b_n^2) \quad .$$

Proceeding in an analogous manner to that in Theorem 4.1, one obtains

$$E[\|Y_n\|^2] \le \sigma_1^2 + (\sigma_2^2 + \lambda^{*2})\, b_n^2 \quad .$$

Therefore,

$$c_{n+1}^2 \le c_n^2 - \Gamma_1(\varepsilon) b_n^2 + \Gamma_2(\varepsilon) \tag{5.16}$$

where

$$\Gamma_1(\varepsilon) = 2\mu\left(\lambda_* - \frac{\lambda^{*2} B^2}{2\varepsilon}\right) - \mu^2(\lambda^{*2} + \sigma_2^2)$$

$$\Gamma_2(\varepsilon) = \mu\varepsilon + \mu^2\sigma_1^2 \quad .$$

Note that $\Gamma_1(\varepsilon) \ge 0$ provided that

$$\varepsilon \ge \frac{\lambda^{*2} B^2}{2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)} \triangleq \varepsilon_0(\mu) \ , \tag{5.15}$$

and

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \ .$$

To bound (5.16) in terms of c_n^2, use the Minkowski inequality to conclude

$$c_n - B \le b_n \le c_n + B \ .$$

From this,

$$c_{n+1}^2 \le [1 - \Gamma_1(\varepsilon)]c_n^2 + 2B\Gamma_1(\varepsilon)c_n - \Gamma_1(\varepsilon)B^2 + \Gamma_2(\varepsilon) \ . \tag{5.17}$$

From (5.17) conclude

$$\limsup_{n\to\infty} c_n \le B + \sqrt{\frac{\Gamma_2(\varepsilon)}{\Gamma_1(\varepsilon)}} \quad .$$

Hence, again by the Minkowski inequality, one obtains

$$\limsup_{n\to\infty} b_n \le 2B + \sqrt{\frac{\Gamma_2(\varepsilon)}{\Gamma_1(\varepsilon)}} \quad .$$

Since $\varepsilon \geq \varepsilon_0$ was arbitrary, one can conclude

$$\limsup_{n\to\infty} b_n \leq 2B + \min_{\varepsilon \geq \varepsilon_0} \sqrt{\frac{\Gamma_2(\varepsilon)}{\Gamma_1(\varepsilon)}} \quad . \tag{5.14}$$

This completes the proof of Theorem 5.3.

The following corollary gives a looser, but more tractable bound.

Corollary 5.3.1. Under the hypothesis of Theorem 5.3

$$\limsup_{n\to\infty} b_n \leq 2B\left(1 + \sqrt{\frac{\lambda^{*2}}{2[2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)]} + \frac{\mu\sigma_1^2/B^2}{2[2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)]}}\right) \tag{5.18}$$

and

$$\lim_{\mu\to 0} \limsup_{n\to\infty} b_n \leq 2B\left(1 + \frac{\lambda^*}{2\lambda_*}\right) \tag{5.19}$$

Proof of Corollary 5.3.1.

Let $\varepsilon = 2\varepsilon_0$ in the original bound for $\limsup_{n\to\infty} b_n$ found in the proof of Theorem 5.3. After a little algebra, the desired result follows.

This completes the proof of Corollary 5.3.1.

The importance of the results given in this section is that they guarantee, in the mean-norm-squared sense, that if the optimum weight vector sequence $\{W_n^*\}$ remains bounded about a vector W_0^*, then the algorithm (3.1) will yield estimates within a finite region about the true minima.

VI. AN EXAMPLE

The purpose of the example discussed in this chapter is to compare the theoretical bound (5.6) with an experimental result obtained by using the LMS adaptation algorithm [42],

$$w_{k+1} = w_k - \mu e_k x_k \ , \tag{6.1}$$

where

$$e_k = d_k - w_k x_k \ .$$

As shown in Appendix E, the LMS adaptation algorithm is a special case of algorithm (3.5) with $G_n(W) = W$ and parameters

$$\lambda_* = \inf_n \lambda_{min}(R_n) \ , \tag{6.2}$$

$$\lambda_* = \sup_n \lambda_{max}(R_n) \ , \tag{6.3}$$

$$\sigma_1^2 = \sup_n E[\| (X_n X_n^T W^* - d_n X_n) \|^2] \ , \tag{6.4}$$

and

$$\sigma_2^2 = \sup_n E[\| X_n X_n^T - R_n \|^2] \ , \tag{6.5}$$

where $\lambda_{min}(R_n)$ and $\lambda_{max}(R_n)$ are the minimum and maximum eigenvalues of R_n, respectively.

The criterion used to measure the performance of the adaptive filter is the excess mean-squared error C_n^2 as defined by (3.14). Using (3.15) and (5.6), on has the bound

$$\lim_{n\to\infty} \sup c_n^2 \leq \lambda^* \left[\frac{\Delta}{1-\alpha^2} + \sqrt{\left(\frac{\Delta}{1-\alpha^2}\right)^2 + \frac{\mu^2\sigma_1^2 - \Delta^2}{1-\alpha^2}} \right] \tag{6.6}$$

where the parameters are as defined in Chapter V.

The object of the adaptive filter w_k in this example was to predict a random process $\{x_k\}$ one time-delay ahead using only the previous value, i.e.,

$$d_k = x_{k+1}$$

and

$$y_k = w_k x_k \quad .$$

The data was generated according to the one-point autoregressive scheme

$$x_k = a_k x_{k-1} + V_k$$

where

x_k = input data value at time k

$a_k = 0.2 \sin \frac{k}{100} + c$

V_k = white, stationary, Gaussian random process with zero mean and unit variance.

The instantaneous error squared, $(x_{k+1} - w_k x_k)^2$, was averaged over 700 points for each value of μ used in the adaptation algorithm (6.1). Two experiments were conducted, one with $c = 0.0$ and the other with $c = 0.5$. The resulting averages are shown as a function of μ by the solid line in Figs. 6.1 and 6.2. The corresponding theoretical bounds are given by the dashed line in the figures.

The discrepancy between the two curves can be accounted for by the following three observations. First, as pointed out in the previous chapter, the theoretical bound is a

$$a_k = 0.2 \sin \frac{k}{100} + c$$

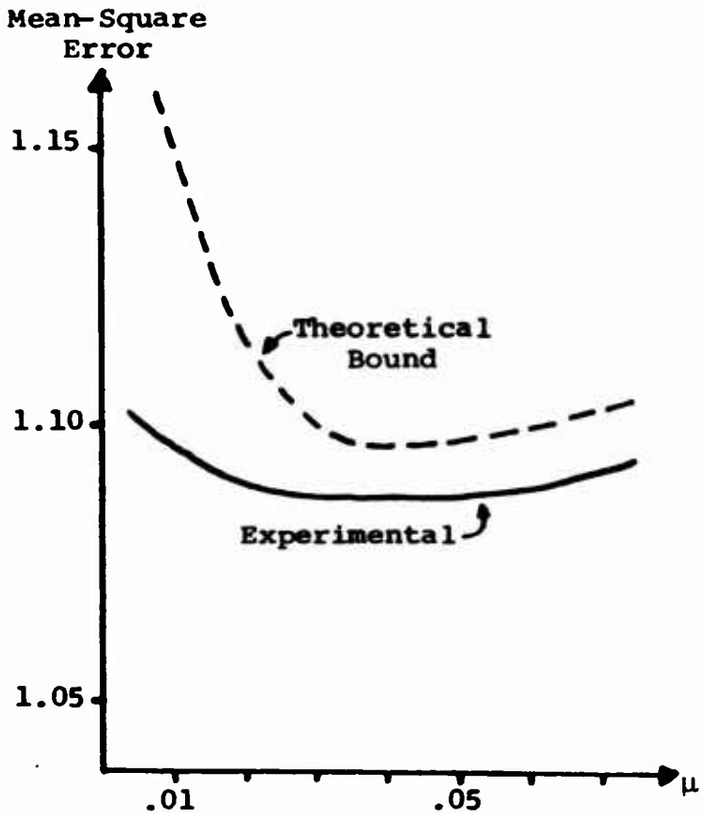


Fig. 6.1 c = 0.0

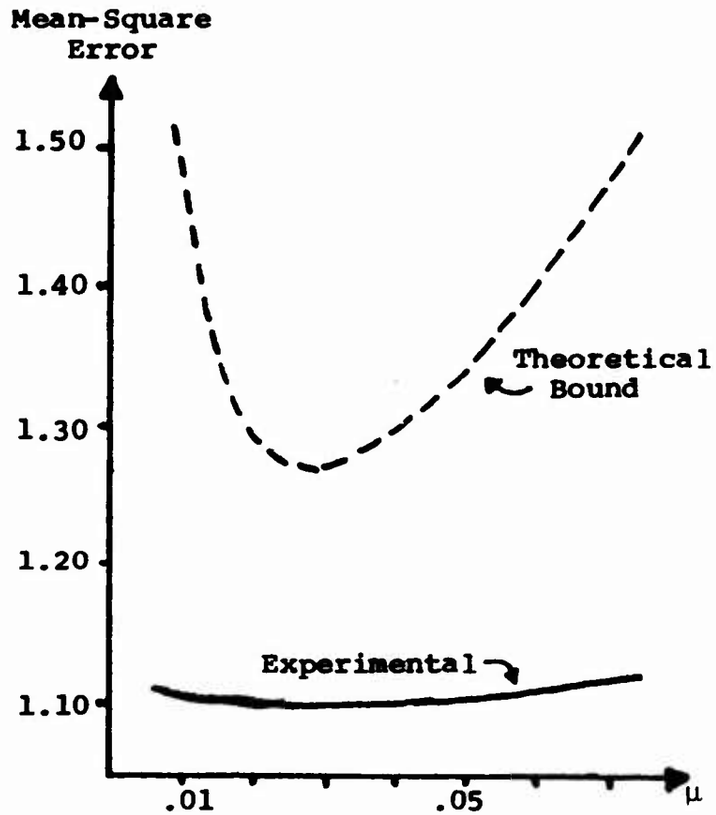


Fig. 6.2 c = 0.5

Theoretical and Experimental Mean-Square Error Curves for Time-Varying Autoregressive Data

worst-case analysis. In other words, the periodic nature of the time-variability of the data is not exploited by the bound. Second, the experimental curve corresponds to the average mean-square error over a period while the bound corresponds to the maximum mean-square error during the period. Last, the sequence of data pairs $\{(d_k, x_k)\}$ is not an independent sequence.

The reason for the larger discrepancy in Fig. 6.2 than in Fig. 6.1 is due, in part, to the increase in the ratio λ^*/λ_* and its effect on the bound (6.5). For the case $c = 0.0$ the ratio is given by

$$\frac{\lambda^*}{\lambda_*} = 1.04$$

while for $c = 0.5$, the ratio becomes

$$\frac{\lambda^*}{\lambda_*} = 1.78 \ .$$

VII. CONCLUSION

A. SUMMARY OF RESULTS

The research reported herein dealt with the convergence properties of stochastic gradient descent algorithms as applied to the sequential adaptation of array processors. In Chapter IV, sufficient conditions were derived for the convergence of these algorithms. A new convergence theorem and proof for certain stochastic approximation algorithms ($\mu_n \to 0$) were presented. These results serve as a guide for deciding whether to use a constant μ or a decreasing μ_n when the dynamics of the nonstationarity are known. However, in general, one has incomplete a priori information concerning the type of nonstationary environment to be encountered. For this reason the worst-case analysis presented in Chapter V is particularly informative as to the type of behavior to expect of the algorithm in general.

Representative curves for the bounds derived in Chapter V for the three types of nonstationarities considered there are shown in Fig. 7.1. It should be emphasized that these bounds are of a worst-case nature. They are summarized by the three theorems:

Theorem 5.1 (Bounded Increment) If

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \quad \text{and} \quad \|W_{n+1}^* - W_n^*\| \leq \Delta$$

then

$$\limsup_{n\to\infty} E[\|W_n - W_n^*\|^2] \leq \left[\frac{\Delta + \mu\sigma_1}{1 - \sqrt{1 - 2\mu\lambda_* + \mu^2(\lambda^{*2} + \sigma_2^2)}}\right]^2 . \tag{5.4}$$

Corollary 5.1.1. (Bounded Variation) If

$$0 < \mu < \frac{2\lambda_*}{\lambda^{*2} + \sigma_2^2} \quad \text{and} \quad \sum_k \Delta_k < \infty ,$$

then

$$\limsup_{n\to\infty} E[\|W_n - W_n^*\|^2] \leq \frac{\mu\sigma_1^2}{2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)} . \tag{5.6}$$

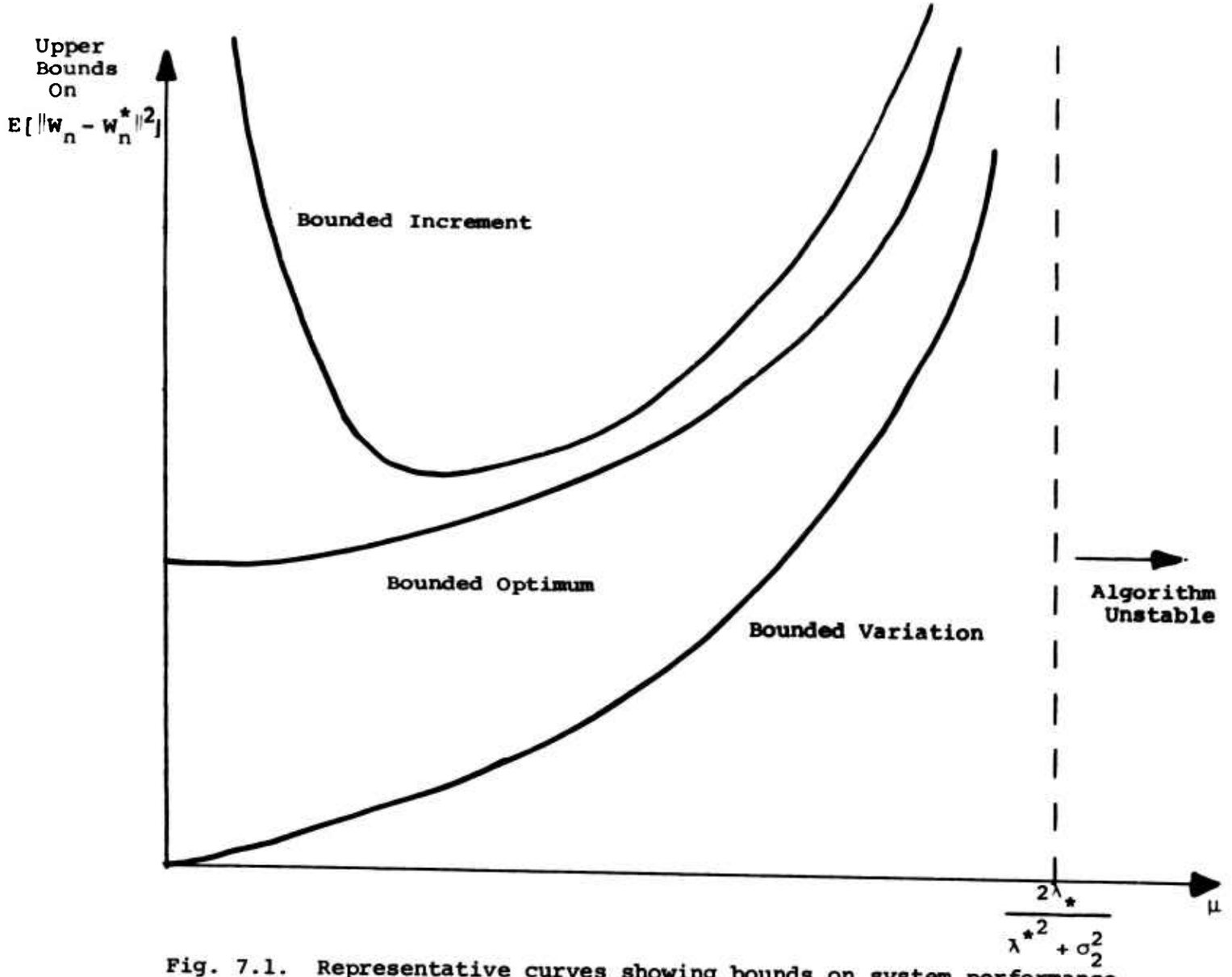


Fig. 7.1. Representative curves showing bounds on system performance.

Theorem 5.3 (Bounded Optimum) If

$$0 < \mu < \frac{2\lambda_*}{\lambda^* + \sigma_2^2}$$

and there exists some weight vector W_o^* such that $\|W_n^* - W_o^*\| \leq B$, then

$$\limsup_{n\to\infty} E[\|W_n - W_n^*\|^2] \leq \left[2B + \min_{\epsilon \geq \epsilon_o} \sqrt{\frac{\epsilon + \mu\sigma_1^2}{2(\lambda_* - \frac{\lambda^{*2}B^2}{2\epsilon}) - \mu(\lambda^{*2} + \sigma_2^2)}} \right]^2 \tag{5.14}$$

$$\leq \left[2B + \frac{2\lambda^* B}{2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)} + \sqrt{\frac{2\mu\sigma_1^2}{2\lambda_* - \mu(\sigma_2^2 + \lambda^{*2})}} \right]^2 ,$$

where

$$\epsilon_o = \frac{\lambda^{*2}B^2}{2\lambda_* - \mu(\lambda^{*2} + \sigma_2^2)} .$$

It should also be noted that the range of values that μ can take in all three cases is independent of the conditions placed on the optimum weight vector sequence $\{W_n^*\}$.

B. OTHER APPLICATIONS OF THE ALGORITHM

The algorithm developed in this research could equally well be applied to a number of problems in the system sciences such as system identification, process control, and pattern recognition when the underlying statistics are alllowed to vary in time. A large number of authors [25] - [30] have considered the application of stochastic approximation theory to these problems in the stationary case, but rather limited consideration has been given to the time-varying problem [23] - [24].

C. RECOMMENDATIONS FOR FURTHER WORK

The following problem areas have been suggested by the research reported in this paper:

i) Although a mechanism has been demonstrated for the behavior of the adaptive processor in non-stationary environments, an optimum choice for the gain constant μ is impossible unless reasonably complete *a priori* knowledge of the nature of the time-variability is available. A procedure for automatically adjusting μ is desirable. An original algorithm, based on the method of steepest descent, for adapting μ is found in Appendix F. Even though the procedure has been shown to work well experimentally, a theoretical proof of convergence would be desirable. (The corresponding deterministic algorithm is discussed in Appendix G.)

ii) The analysis presented in this research did not exploit the linearity property of the gradient $J_n(W)$. By incorporating this additional property into the analysis, one might be able to obtain tighter bounds on the system performance. As an indication of how this might be done, see Appendix H for a discussion of the scalar problem.

iii) Much attention has been given to comparing stochastic approximation algorithms and the Kalman-Bucy filter [55] - [57]. The analysis presented here enlarges upon the problems for which the two methods can be compared. A further elaboration on this topic could prove fruitful, since the algorithm (3.5) is computationally simpler to implement than the corresponding Kalman-Bucy filter. For a further discussion, see Appendix I.

APPENDIX A

A LEMMA ON THE LIMIT SUPERIOR FOR A SEQUENCE OF NON-NEGATIVE REAL NUMBERS

The following lemma establishes an important algebraic property of the limit superior of a sequence of non-negative real numbers.

Lemma. Let $\{x_n\}$ be a non-negative sequence of real numbers. Define

$$x^* = \limsup_{n \to \infty} x_n$$

$$(x^2)^* = \limsup_{n \to \infty} x_n^2$$

then

$$(x^2)^* = (x^*)^2$$

Proof of the Lemma.

Assume the contrary. Let $(x^2)^* < (x^*)^2$. Let $a > 0$ be such that $(x^2)^* < a^2 < (x^*)^2$. Now there exists an N_1 such that for all $n \geq N_1$ it follows $x_n^2 \leq (x^2)^* + \epsilon_1$, where we let $\epsilon_1 = a^2 - (x^2)^*$. This implies $x_n \leq a$. But for all $\epsilon_2 > 0$, it is the case that $x_n > x^* - \epsilon_2$ infinitely often. Let $\epsilon_2 = x^* - a$. Then $x_n > a$ infinitely often. Hence, a contradiction. Now suppose that $(x^*)^2 < (x^2)^*$. Let $a > 0$ be such that $(x^*)^2 < a^2 < (x^2)^*$. Since there exists an N_2 such that for all $n \geq N_2$ we

have $x_n \leq x^* + \epsilon_3$ where $\epsilon_3 = a - x^*$. Thus, $x_n \leq a$. But for all $\epsilon_4 > 0$ we have $x_n^2 > (x^2)^* - \epsilon_4$ infinitely often. Letting $\epsilon_4 = (x^2)^* - a^2$ we have $x_n^2 > a^2$. Again, a contradiction. Thus, it must be the case that $(x^2)^* = (x^*)^2$.

This completes the Proof of the Lemma.

APPENDIX B

PROOF OF KRONECKER'S LEMMA

This appendix presents the proof of Kronecker's lemma stated in Chapter IV. The lemma is:

Lemma 4.2.1 (Kronecker) Let $\{x_k\}$ be a sequence of real numbers. Let $\{a_k\}$ be a sequence of positive numbers converging monotonically upward to infinity. If $\sum_{k=1}^{n} \frac{x_k}{a_k} = s_n$ converges to some finite number, say s, then

$$\lim_{n\to\infty} \frac{1}{a_n} \sum_{k=1}^{n} x_k = 0 .$$

Proof of Lemma 4.2.1.

Before proving this lemma we need Abel's lemma on partial summation:

Lemma (Abel). Let $\{y_n\}$ and $\{z_n\}$ be sequences. Define $s_n = \sum_{k=1}^{n} y_k$. If $m > n$, then

$$\sum_{j=n}^{m} y_j z_j = (z_m s_m - z_n s_{n-1}) + \sum_{j=n}^{m-1} s_j (z_j - z_{j+1}) .$$

Proof of Abel's Lemma.

Noting that $y_j = s_j - s_{j-1}$, we may write

$$\sum_{j=n}^{m} y_j z_j = \sum_{j=n}^{m} z_j (s_j - s_{j-1})$$

$$= \sum_{j=n}^{m} z_j s_j - \sum_{j=n}^{m} z_j s_{j-1}$$

$$= z_m s_m + \sum_{j=n}^{m-1} z_j s_j - \sum_{j=n}^{m-1} z_{j+1} s_j - z_n s_{n-1}$$

$$= (z_m s_m - z_n s_{n-1}) + \sum_{j=n}^{m-1} s_j (z_j - z_{j+1}) .$$

This completes the proof of Abel's lemma.

Defining $y_j = \frac{x_j}{a_j}$, $z_j = a_j$, $s_0 = 0$, $a_0 = 0$, we have by Abel's lemma

$$\sum_{j=1}^{n} x_j = a_n s_n + \sum_{j=0}^{n-1} s_j (a_j - a_{j+1}) .$$

Using the identity

$$\frac{1}{a_n} \sum_{j=0}^{n-1} (a_{j+1} - a_j) = 1$$

we may write

$$\frac{1}{a_n} \sum_{j=1}^{n} x_j = \frac{1}{a_n} \sum_{j=0}^{n-1} (s_n - s_j)(a_{j+1} - a_j) .$$

To show $\frac{1}{a_n} \sum_{j=1}^{n} x_j$ converges to zero, we use the repeated application of the triangle inequality to obtain

$$\left| \frac{1}{a_n} \sum_{j=1}^{n} x_j \right| \leq \frac{1}{a_n} \sum_{j=0}^{n-1} |s_n - s_j| (a_{j+1} - a_j) .$$

Since s_n converges to some finite number s, there exists some integer N_1 such that for all $n,m > N_1$ we have $|s_n - s_m| < \varepsilon/2$. Therefore, for $n > N_1$,

$$\left|\frac{1}{a_n}\sum_{j=0}^{n} x_j\right| \leq \frac{1}{a_n}\sum_{j=0}^{N_1-1} |s_n - s_j|(a_{j+1} - a_j) + \frac{\varepsilon}{2a_n}\sum_{j=N_1}^{n}(a_{j+1} - a_j)$$

$$\leq \frac{1}{a_n}\sum_{j=0}^{N_1-1} |s_n - s_j|(a_{j+1} - a_j) + \frac{\varepsilon}{2}.$$

Since a_n converges monotonically to infinity, there exists an $N_2 > N_1$, such that for $n \geq N_2$

$$\frac{1}{a_n}\sum_{j=0}^{N_1-1} |s_n - s_j|(a_{j+1} - a_j) < \varepsilon/2.$$

Therefore, for $n \geq N_2$

$$\left|\frac{1}{a_n}\sum_{j=1}^{n} x_j\right| \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2}\frac{1}{a_n}\sum_{j=N_1}^{n}(a_{j+1} - a_j)$$

$$\leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} = \varepsilon.$$

Hence, for sufficiently large n we can make $\frac{1}{a_n}\sum_{j=1}^{n} x_j$ arbitrarily small.

<u>This completes the proof of Kronecker's lemma.</u>

APPENDIX C

A MARTINGALE CONVERGENCE THEOREM

The purpose of this appendix is to prove the following lemma, needed in the proof of Theorem 4.2.

Lemma 4.2.2. Let $(\Omega, \mathcal{F}, P)$ be a probability space. Let $\{X_n, \mathcal{F}_n\}$ be a stochastic sequence on $(\Omega, \mathcal{F}, P)$ with $\mathcal{F}_n \subseteq \mathcal{F}_{n+1}$. Let $\{a_k\}$ be a sequence of non-negative real numbers. Assume the following conditions hold

(H1) $\sup_k E[|X_k|] < \infty$

(H2) $\sum_1^\infty a_k < \infty$

(H3) $E[X_{k+1} | \mathcal{F}_k] \geq X_k - a_k$ for all k a.e.

Then,

$$\lim_{n\to\infty} X_n = X_\infty \quad \text{where} \quad E[|X_\infty|] < \infty \quad \text{a.e.}$$

Remark: If $a_k = 0$ for all k, then by the basic martingale convergence theorem [61], the above conclusion follows. The effect of the a_k is to translate the X_k. Since $\sum_1^\infty a_k < \infty$, one should expect the above lemma. We now formalize this observation.

Proof of the Lemma.

Define $Z_k = X_k - \sum_{\ell=k}^\infty a_\ell$. Note that $\{Z_k, \mathcal{F}_k\}$ is a submartingale since

$$E[Z_{k+1}|\mathcal{F}_k] = E\left[X_{k+1} - \sum_{\ell=k+1}^{\infty} a_\ell \,\middle|\, \mathcal{F}_k\right]$$

$$= E[X_{k+1}|\mathcal{F}_k] - \sum_{\ell=k+1}^{\infty} a_\ell$$

$$\geq X_k - \sum_{\ell=k}^{\infty} a_\ell = Z_k \ .$$

It follows immediately that

$$E[|Z_k|] \leq E[|X_k|] + \sum_{\ell=k}^{\infty} a_\ell \leq \sup_k E[|X_k|] + \sum_{\ell=1}^{\infty} a_\ell$$

and by the martingale convergence theorem [61],

$$\lim_{k\to\infty} Z_k = Z_\infty \quad \text{and} \quad E[|Z_\infty|] < \infty \quad \text{a.e.}$$

Since $\sum_{\ell=k}^{\infty} a_\ell$ converges by the monotone convergence theorem, it must be the case that

$$\lim_{k\to\infty} X_k = X_\infty \quad \text{a.e.}$$

Moreover, since $X_\infty = Z_\infty + \sum_{k=1}^{\infty} a_k$, $E[|X_\infty|] < \infty$.

This completes the proof of the lemma.

APPENDIX D

AN INEQUALITY BETWEEN THE ABSOLUTE MOMENTS ABOUT ZERO OF ORDER 1 AND ORDER 2

An inequality useful in proving convergence theorems is

Lemma. Let x be a random variable on some probability space $(\Omega, \mathcal{F}, P)$. Then, for all $\epsilon > 0$

$$E[|x|] \leq \frac{\epsilon}{2} + \frac{E[|x|^2]}{2\epsilon}$$

Proof of the Lemma.

The trick is to note that for all $a > 0$ and $\epsilon > 0$

$$\tfrac{1}{2}\left(\frac{a}{\epsilon} + \frac{\epsilon}{a} \right) \geq 1$$

Letting $a = \{E[|x|^2]\}^{\frac{1}{2}}$, we can obtain

$$\tfrac{1}{2}\left(\epsilon + \frac{1}{\epsilon} E[|x|^2]\right) \geq \{E[|x|^2]\}^{\frac{1}{2}} .$$

Applying the Cauchy-Schwarz inequality ($p = 2$ in Holder's inequality) we obtain the desired result,

$$\frac{\epsilon}{2} + \frac{E[|x|^2]}{2\epsilon} \geq E[|x|] .$$

This completes the proof of the lemma.

APPENDIX E

SOME ADAPTATION ALGORITHMS

Examples of some specific adaptation algorithms which are members of the class of iterative procedures defined by (3.5) will be given in this appendix.

A. A STOCHASTIC APPROXIMATION ALGORITHM

If the statistics of the filtering problem are wide-sense stationary (i.e., $R_n = R$ and $P_n = P$ for all n) the stochastic approximation algorithm suggested by Gardner [21] for estimating W^* based on the data set

$$\{(d_k, X_k) : k = 1, 2, \ldots n\} \triangleq \{(d_k, X_k)\}_1^n ,$$

is given by

$$W_{n+1} = W_n + \mu_n e_n X_n , \qquad \text{(E.1)}$$

where

W_n is the estimate of W^* based on $\{(d_k, X_k)\}_1^{n-1}$

$e_n = d_n - W_n^T X_n$ = the error between the desired filter output and actual filter output at time n

μ_n = gain or weighting constant at time n .

For the stationary problem, the model for the optimum weight vector given by (3.3) becomes

$$W_{n+1}^* = F_n(W_n^*) ,$$

where for all W

$$F_n(W) = W \ ;$$

i.e., for all n

$$W^*_{n+1} = W^*_1 \triangleq W^* \ .$$

Thus, the algorithm (E.1) is of the correct form to apply Theorem 4.2 to show convergence of the sequence $\{W_n\}$ to W^*. All that one has to do is verify that $Y_n \triangleq -e_n X_n$ satisfies (3.9) and (3.10).

The conditions under which Y_n satisfies (3.9) and (3.10) will be derived in Section C. Both the Gaussian and non-Gaussian cases will be considered there.

B. CONSTANT-μ ALGORITHMS

If the statistics of the problem are not stationary, the algorithm given by Widrow [42] for estimating W^*_{n+1} based on the data set $\{(d_k, X_k)\}_1^n$ is

$$W_{n+1} = W_n + \mu e_n X_n \qquad \text{(E.2)}$$

where

W_n is the estimate of W^*_n based on $\{(d_k, X_k)\}_1^{n-1}$

$e_n = d_n - W_n^T X_n$ = the error between the desired output and actual output at time n

μ = gain or weighting constant.

Although this algorithm is similar to (E.1), note that the gain constant is no longer a function of time. This enables the LMS algorithm (E.2) to track a rather general sequence of weight vectors $\{W_n^*\}$. (See Chapter V.)

In many applications, the desired response sequence $\{d_n\}$ is not available. This is the case, for example, with the filtering problem, where the desired response is the unknown signal. However, if the correlation matrix P_n is known, Griffiths [44] has suggested the modified algorithm

$$W_{n+1} = W_n - \mu(y_n X_n - P_n) \qquad (E.3)$$

where

$$y_n = W_n^T X_n = \text{output of adaptive filter.}$$

With this algorithm, one does not need the desired response to be able to adjust the filter.

Widrow, et al. [43] have proposed an alternate procedure for supplying training signals while simultaneously processing the received signal. Griffiths [44] showed this approach is equivalent to his algorithm,

$$W_{n+1} = W_n - \mu[X_n X_n^T W_n + \beta^2 C(C^T W_n - V)] \ , \qquad (E.4)$$

where the matrix C is related to the spatial characteristics of the array and V controls the temporal characteristics.

By allowing β^2 to become large, the sequence $\{W_n\}$ converges in the mean-value to the maximum-likelihood weight vector

$$W^* = R^{-1}C(C^TR^{-1}C)^{-1}V \ .$$

Kelly [7] has shown that when the data pairs (d_k, x_k) are jointly Gaussian, the maximum-likelihood weight vector is also the weight vector which minimizes output power $E[y_n^2]$ subject to the linear constraint $C^TW = V$. Rosen [10] has developed a gradient projection method for iteratively computing this constrained weight vector. Lacoss [40] has extended the technique to the stochastic design problem. Frost [48] has modified this procedure for implementation on a digital computer. Frost's procedure automatically corrects for quantization errors introduced in the constraint equation during adaptation.

The algorithm suggested by Frost is

$$W_{n+1} = P(W_n - \mu y_n X_n) + Q \qquad \text{(E.5)}$$

where

$$P = I - C(C^TC)^{-1}C^T$$

$$Q = C(C^TC)^{-1}V,$$

and the optimum weight vector is given by

$$W^* = R^{-1}C(C^TR^{-1}C)^{-1}V \ .$$

Mathematically, algorithm (E.5) is equivalent to the algorithm

$$W_{n+1} = P(W_n - \mu y_n X_n + \mu RW^*) + Q \qquad \text{(E.6)}$$

since $PRW^* = 0$. This algorithm satisfies (3.5) with

$$G_n(W) \triangleq PW + Q \ .$$

Note also that W^* satisfies

$$W^* = PW^* + Q \ .$$

Hence, the model (3.3) applies with

$$F_n(W) \triangleq PW + Q \ ,$$

and

$$U_n \equiv 0 \ .$$

Convergence of (E.6) implies convergence of (E.5).

The convergence analysis of the algorithms given in this section for the stationary statistics problem follows from Corollary 4.1.1 where the relevant parameters are found in a fashion similar to that done in Section C for the stochastic approximation algorithm (E.1). The behavior of (E.2 - E.6) for the nonstationary problem may be obtained by reference to the results in Chapter V. It should be noted that these results are far more general than any previous convergence analysis for (E.2 - E.6) [42] - [48].

C. Sufficient Conditions for Convergence of Stochastic Approximation Algorithm

1. General Non-Gaussian Case

To verify (3.9), note that, by the independent samples assumption for $\{(d_k, X_k)\}$, W_n and (d_n, X_n) are independent. Hence,

$$\begin{aligned} E[Y_n | W_n, W^*] &= E[X_n X_n^T W_n | W_n, W^*] - E[d_n X_n | W_n, W^*] \\ &= E[X_n X_n^T] W_n - E[d_n X_n] \\ &= RW_n - RW^* \\ &= J(W_n) \ . \end{aligned}$$

To verify (3.19) use the chain of inequalities,

$$\begin{aligned} &\| (X_n X_n^T - R)(W_n - W^*) - (X_n X_n^T W^* - d_n X_n) \|^2 \\ &\quad \leq [\| (X_n X_n^T - R)(W_n - W^*) \| + \| (X_n X_n^T W^* - d_n X_n) \|]^2 \\ &\quad \leq 2 \| (X_n X_n^T - R) \|^2 \| W_n - W^* \|^2 + 2 \| (X_n X_n^T W^* - d_n X_n) \|^2 , \end{aligned}$$

to show that by the independent samples assumption

$$E[\| Y_n - J(W_n) \|^2 | W_n, W^*] \leq \sigma_1^2 + \sigma_2^2 \| W_n - W^* \|^2 ,$$

where

$$\sigma_1^2 = 2E[\| (X_n X_n^T W^* - d_n X_n) \|^2]$$

and

$$\sigma_2^2 = 2E[\| (X_n X_n^T - R) \|^2] \ .$$

Therefore, by Theorem 4.2, under the independent data pairs assumption and existing moments of order four, the $\{W_n\}$ given by (E.1) converge with probability one and in mean-square to W^* if $\{\mu_n\}$ satisfy the usual conditions.

$$\text{(i)} \quad \mu_n \geq 0$$
$$\text{(ii)} \quad \Sigma \mu_n = \infty$$
$$\text{(iii)} \quad \Sigma \mu_n^2 < \infty \ .$$

2. Gaussian Case

If, in addition to the assumptions given in part 1, one assumes that the data pair (d_n, x_n) is jointly Gaussian, he can obtain tighter bounds for σ_1^2 and σ_2^2. This is shown by the following argument.

By the independent samples assumption it follows that

$$E[\| (X_n X_n^T - R)(W_n - W^*) \|^2 | W_n, W^*]$$
$$= (W_n - W^*)^T E[(X_n X_n^T - R)^2] (W_n - W^*) \ .$$

Senne [46] has shown that if X_n is Gaussian, then

$$E[(X_n X_n^T - R)^2] = R^2 + R \ tr\{R\} \ ,$$

where

$$tr\{R\} \triangleq \sum_{i=1}^{p} r_{ii} = \text{trace of } R \ .$$

Therefore,

$$E[\| (X_n X_n^T - R)(W_n - W^*) \|^2 | W_n, W^*] \leq \lambda_{max}(\lambda_{max} + tr\{R\}) \|W_n - W^*\|^2 \ ,$$

where λ_{max} is the maximum eigenvalue of R.

Define

$$\varepsilon_n^* = d_n - X_n^T W^* .$$

Since the data pair (d_n, X_n) are jointly Gaussian, it follows that any linear combination is Gaussian [ ]. In particular, ε_n^* is Gaussian. Moreover, since by definition of W^*,

$$E[\varepsilon_n^* X_n] = 0 = E[\varepsilon_n^*] E[X_n] ,$$

it follows [ ] that ε_n^* and X_n are independent. Hence, ε_n^* and any function of X_n are independent [ ]. Therefore,

$$E[(X_n X_n^T W^* - d_n X_n)^T (X_n X_n^T - R)(W_n - W^*) | W_n, W^*]$$

$$= E[\varepsilon_n^*] E[X_n^T (X_n X_n^T - R)] (W_n - W^*)$$

$$= 0 ,$$

and

$$E(\|X_n X_n^T W^* - d_n X_n)\|^2 | W_n, W^*] = E[(\varepsilon_n^*)^2] E[\|X_n\|^2]$$

$$= (tr\{R\}) \xi(W^*) .$$

Hence,

$$E[\|Y_n - J(W_n)\|^2 | W_n, W^*] \leq \xi(W^*)[tr\{R\}]$$

$$+ (\lambda_{max}^2 + \lambda_{max} tr\{R\}) \|W_n - W^*\|^2 .$$

APPENDIX F

TIME-VARYING ADAPTATION ALGORITHM

One of the major problems connected with using the LMS adaptation algorithm,

$$W_{n+1} = W_n + \mu e_n X_n \quad , \tag{3.5}$$

discussed in Appendix E is the choice of μ to use during adaptation. Without any a priori knowledge of the time-varying characteristics of the data, one would like an algorithm which automatically seeks out the optimum value of μ without resorting to a random trial-and-error method. The following algorithm was originally suggested by the author to accomplish this task:

$$W_{n+1} = W_n + \mu_n e_n X_n \tag{F.1}$$

$$\mu_n = \mu_{n-1} + \lambda e_{n-1} e_n X_{n-1}^T X_n \tag{F.2}$$

The reasoning behind the above scheme is as follows. Consider (F.1) without regard to how μ_n is chosen. Since e_{n+1}^2 is a function of W_{n+1}, which in turn depends on the sequence $\{\mu_i\}$, one would be led to pick μ_{n+1} according to

$$\mu_{n+1} = \mu_n - \frac{\lambda}{2} \frac{\partial e_{n+1}^2}{\partial \mu_n}$$

where $\frac{\partial e_{n+1}^2}{\partial \mu_n}$ is the gradient of e_{n+1}^2 with respect to μ_n. Evaluating the gradient,

$$\frac{\partial}{\partial \mu_n} e_{n+1}^2 = 2e_{n+1} \frac{\partial e_{n+1}}{\partial \mu_n}$$

$$= 2e_{n+1} (d_{n+1} - X_{n+1}^T W_{n+1})$$

$$= -2e_{n+1} X_{n+1}^T \left(\frac{\partial}{\partial \mu_n} W_{n+1} \right)$$

$$= -2e_{n+1} X_{n+1}^T \frac{\partial}{\partial \mu_n} (W_n - \mu_n e_n X_n)$$

$$= -2e_n e_{n+1} X_{n+1}^T X_n$$

we obtain the algorithm

$$\mu_{n+1} = \mu_n + \lambda e_n e_{n+1} X_n^T X_{n+1}$$

While most of the time this algorithm did perform quite well under experimental conditions, there were instances for which it did diverge. The reason for this behavior is that the value of λ to be used depends upon the initial weight vector W_1. This is demonstrated by the following example.

Example.

The deterministic adaptation algorithm equivalent to (F.1) and (F.2) is given by

$$\mu_n = \mu_{n-1} + \lambda Z^T(n-1) Z(n) \qquad \text{(F.3)}$$

$$Z(n+1) = [I - \mu_n R] Z(n) \qquad \text{(F.4)}$$

where $Z(n)$ is related to $W(n)$ by

$$Z(n) = R(W(n) - W^*)$$

R = covariance matrix of the input data (which is assumed to be stationary for this example)

W^* = optimum finite-dimensional linear estimator.

We shall consider the case with

$$Z(n) = \begin{bmatrix} z_1(n) \\ z_2(n) \end{bmatrix} \qquad R = \begin{bmatrix} r & 0 \\ 0 & r \end{bmatrix} \qquad (r > 0)$$

Then, our algorithm becomes

$$\mu_n = \mu_{n-1} + \lambda(1-\mu_{n-1}r)(z_1^2(n-1) + z_2^2(n-1))$$

$$z_i(n+1) = (1 - \mu_n r) z_i(n) \qquad i = 1,2$$

Defining

$$\alpha(n) = 1 - \mu_n r$$

$$\beta = \lambda r$$

we can write the above algorithm in the form

$$\alpha(n) = \alpha(n-1) - \beta\alpha(n-1)(z_1^2(n-1) + z_2^2(n-1)$$

$$z_i(n+1) = \alpha(n) z_i(n) \qquad i = 1,2$$

Letting $z_1(1) = z_2(1)$, we can further simplify to

$$\alpha^2(n) = \alpha^2(n-1)(1 - 2\beta z_1^2(n-1))^2$$

$$z_1^2(n+1) = \alpha^2(n) z_1^2(n)$$

If $z_1^2(1) = \frac{1+\epsilon}{\beta}$ and $\mu_1 = 0$ $(\alpha(1) = 1)$ where $\epsilon > 0$, then

$$\begin{cases} z_1^2(2) = \frac{1+\epsilon}{\beta} \\ \alpha^2(2) = (1+2\epsilon)^2 \end{cases}$$

$$\begin{cases} z_1^2(3) = \frac{1+\epsilon}{\beta}(1+2\epsilon)^2 \\ \alpha^2(3) = (1+2\epsilon)^2[1-2(1+2\epsilon)^2(1+\epsilon)]^2 \end{cases}$$

$$\geq (1+2\epsilon)^3$$

etc.

with $\alpha^2(n) \to \infty$ as $n \to \infty$. Hence $z_1^2(n) \to \infty$ as $n \to \infty$. Thus, the choice of λ (or β above) for stability of the algorithm does depend on initial conditions.

End of Example.

Is it possible to modify the time-varying algorithm such that initial conditions no longer govern the stability of the processor? The argument to be presented for modifying the deterministic gradient procedure suggests that the stability of the corresponding LMS time-varying algorithm is independent of initial conditions. No theoretical proof is available as yet to support this conjecture;† however the experimental results at the conclusion of this argument do support this hypothesis.

An important reason for why the stability of the method of steepest descent algorithm doesn't depend on initial conditions is linearity. Although the modified algorithm can't be made linear, the norm of $\underline{Z}(n)$ can almost be made linear by updating μ_n according to

† For the deterministic gradient procedure when the ratio λ^*/λ_* is less than two, convergence can be proven. See Appendix G for a proof.

$$\mu_n = \mu_{n-1} + \lambda \frac{z^T(n-1)Z(n)}{\|Z(n-1)\|^2}$$

The algorithm then becomes

$$\mu_n = \mu_{n-1} + \lambda \frac{\underline{Z}^T(n-1)[I - \mu_{n-1} R]\ Z(n-1)}{\|Z(n-1)\|^2} \qquad (F.5)$$

$$Z(n+1) = [I - \mu_n R]\ Z(n) \quad . \qquad (F.6)$$

The corresponding stochastic algorithm is

$$\mu_n = \mu_{n-1} + \lambda \frac{e(n)}{e(n-1)} \frac{X^T(n)\,X(n-1)}{\|X(n-1)\|^2} \qquad (F.7)$$

$$W(n+1) = W(n) - \mu_n e(n) X(n) \qquad (F.8)$$

where

$e(n)$ = difference between desired output and actual output of adaptive filter

$X(n)$ = data vector at time n

The stochastic algorithm can be further modified by observing a few properties of the deterministic algorithm, the idea being that we want the stochastic algorithm to behave in as deterministic a way as possible without knowledge of the <u>a priori</u> statistics of the problem. Define

$$\alpha_{n-1} = \frac{Z_{n-1}^T R\ Z(n-1)}{\|Z(n-1)\|^2}$$

and note

$$\lambda_* \leq \alpha_{n-1} \leq \lambda^*$$

where

$$\lambda_* = \text{smallest eigenvalue of } R$$

$$\lambda_* = \text{largest eigenvalue of } R$$

The algorithm for μ_n becomes

$$\mu_n = \mu_{n-1} + \lambda(1 - \alpha_{n-1}\mu_{n-1})$$

If $\mu_{n-1} \geq 0$, we have

$$\lambda_*\mu_{n-1} \leq \alpha_{n-1}\mu_{n-1} \leq \lambda^*\mu_{n-1}$$

For convergence we want

$$0 \leq \lambda_*\mu_{n-1} \leq \lambda^*\mu_{n-1} \leq 2$$

which implies

$$|1 - \alpha_{n-1}\mu_{n-1}| \leq 1$$

by the previous inequality.

This suggests the further modification in the stochastic algorithm:

$$\mu_n = \left[\mu_{n-1} + \lambda\left[-\frac{e(n)}{e(n-1)}\,\frac{X^T(n)\,\underline{X}(n-1)}{\|X(n-1)\|^2}\right]_{-1}^{1}\right]_0^K \qquad \text{(F.9)}$$

and

$$W(n+1) = W(n) - \mu_n e(n)\,X(n) \qquad \text{(F.10)}$$

where

$$[y]_a^b \triangleq \begin{cases} b & y \geq b \\ y & a < y < b \\ a & y \leq a \end{cases}$$

K is some constant $\leq 2/\lambda^*$

Since λ^* is in general unknown, the choice of K would appear to be a problem. However, experimental evidence suggests that the choice of K is relatively unimportant (the results presented here used $K = \infty$). This is not really too surprising, for if μ becomes too large we should expect the quantity

$$\frac{e(n)}{e(n-1)} \frac{X^T(n)X(n-1)}{\|X(n-1)\|^2}$$

to be biased negatively. Thus μ would tend to become smaller.

The following set of experiments have been conducted on the IBM 1130 to determine the behavior of the stochastic algorithm. The data was generated according to the one-point autoregressive scheme

$$x_t = a_{t-1} x_{t-1} + v_t$$

where

x_t = data value at time t

a_t = $b \sin(wt) + c$

v_t = white, stationary, gaussian random variable with variance one and zero mean.

The purpose of the adaptation filter was to predict the process $\{x_t\}$ one-point ahead of time. Figs. F.1 and F.2 show the mean-squre-error of the prediction filter as a function of the initial choice μ_o for μ for various values of λ, b, w, and c. The averages were computed over 700 points after 3800 adaptations. It should be recalled that for $\lambda = 0$, one has the basic LMS adaptation algorithm.

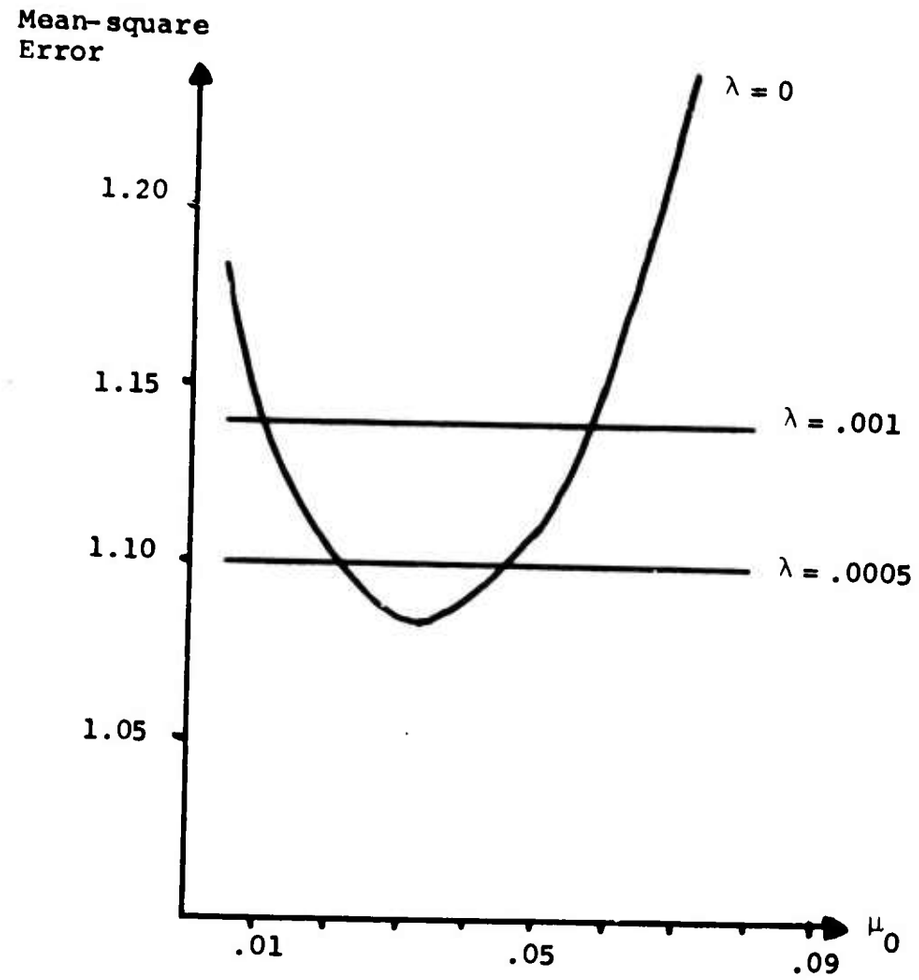


Fig. F.1. Mean-squared Error for Time-varying Algorithm:

b=0.4 c=0.5 $w^{-1}=200$

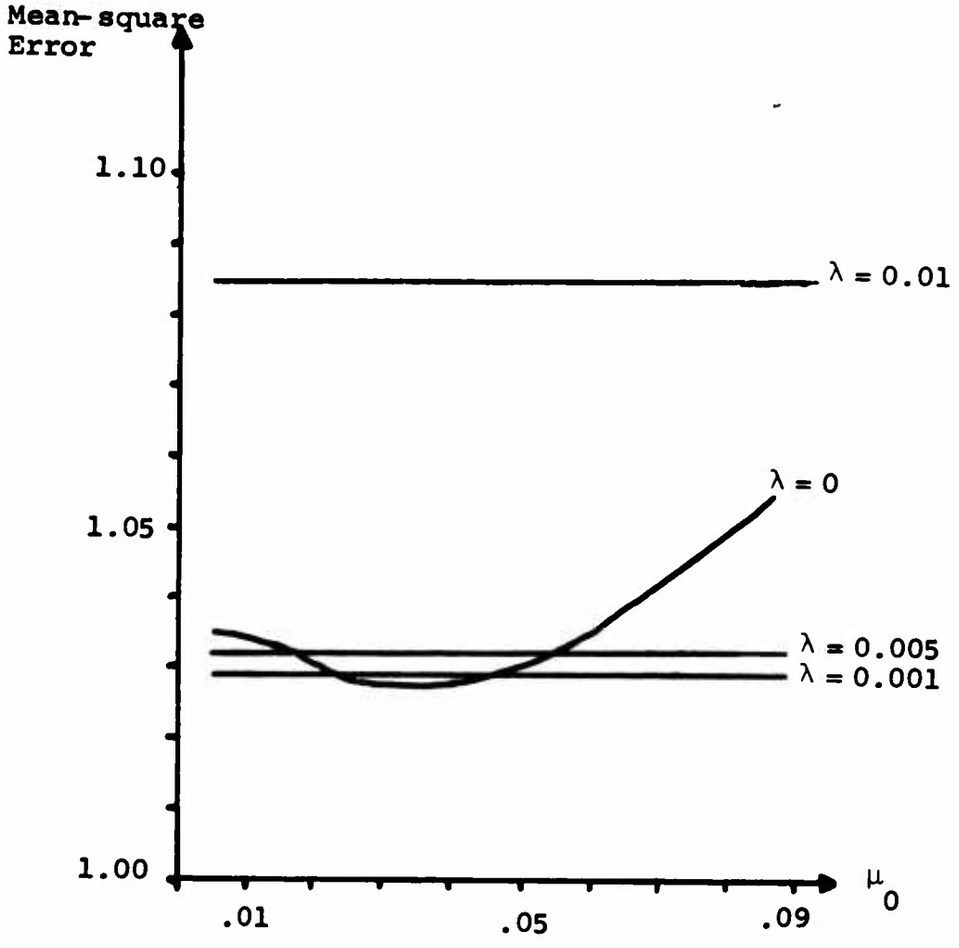


Fig. F.2. Mean-squared Error for Time-varying Algorithm:

b=0.2 c=0.5 $w^{-1}=200$

It is easy to show, just by repeating the argument given for the LMS adaptation algorithm, that the algorithm given by

$$W_{n+1} = W_n - \mu_n Y_n \tag{3.5}$$

can be modified to yield

$$\mu_n = \mu_{n-1} + \lambda \left[\frac{Y_n^T Y_{n-1}}{\|Y_{n-1}\|^2} \right]_{-1}^{+1}$$

Other schemes for varying the step-size are discussed in the references [30] - [32].

APPENDIX G

A DISCUSSION OF THE DETERMINISTIC TIME-VARYING ADAPTATION ALGORITHM OF APPENDIX F

The purpose of this appendix is to discuss the deterministic time-varying adaptation algorithm developed in Appendix F. First, a convergence theorem:

Theorem. Let the sequence of vectors $\{Z_n\}$ be generated by the pair of recursive relations

$$\mu_n = \mu_{n-1} + \lambda \frac{Z_{n-1}^T [I - \mu_{n-1} R] Z_{n-1}}{\|Z_{n-1}\|^2} \qquad (F.5)$$

$$Z_{n+1} = [I - \mu_n R] Z_n \qquad (F.6)$$

where R is a positive-definite, symmetric matrix with eigenvalues lying in the interval $[0 < \lambda_* , \lambda^* < \infty]$. If $\lambda^*/\lambda_* < 2$, $0 < \lambda < 1/\lambda^*$, and $0 \leq \mu_1$, then

$$\lim_{n \to \infty} W_n = W^* .$$

Remark. Since R is positive definite and $Z_n = R(W_n - W^*)$, the convergence of Z_n to 0 is equivalent to

$$\lim_{n \to \infty} W_n = W^*$$

Proof of Theorem.

The first step is to derive sufficient conditions on the sequence $\{\mu_n\}$ for convergence of (F.6). Note that

$$\|Z_{n+1}\| \leq \|I - \mu_n R\| \|Z_n\|$$

$$\leq \left[\prod_{k=N}^{n} \|I - \mu_k R\| \right] \|Z_N\|$$

for some $N \geq 1$. Now $\|I - \mu_n R\| < 1$ if, say, for some $\delta > 0$

$$\delta \leq \mu_n \leq \frac{2}{\lambda^*} - \delta , \qquad \text{(G.1)}$$

for sufficiently large n . Hence.

$$\lim_{n \to \infty} \left[\prod_{k=N}^{n} \|I - \mu_k R\| \right] = 0$$

and consequently,

$$\lim_{n \to \infty} Z_n = 0 .$$

To verify that the μ_n generated according to (F.5) satisfy (G.1) for sufficiently large n , proceed as follows.

Note that (F.5) may be written

$$\mu_n = (1 - \lambda \alpha_{n-1}) \mu_{n-1} + \lambda$$

where

$$\alpha_n = \frac{Z_n^T R Z_n}{\|Z_n\|^2} .$$

Therefore, if $0 \leq \lambda \leq \frac{1}{\lambda^*}$ and $\mu_{n-1} \geq 0$, then

$$0 < (1 - \lambda \lambda^*) \mu_{n-1} + \lambda \leq \mu_n \leq (1 - \lambda \lambda_*) \mu_{n-1} + \lambda . \qquad \text{(G.2)}$$

Consequently, if $\mu_1 \geq 0$, then (G.2) holds for all n .

Upon successive iteration, one obtains

$$0 < (1 - \lambda \lambda^*)^{n-1} (\mu_1 - \frac{1}{\lambda^*}) + \frac{1}{\lambda^*} \leq \mu_n \leq (1 - \lambda \lambda_*)^{n-1} (\mu_1 - \frac{1}{\lambda_*}) + \frac{1}{\lambda_*} .$$

Hence, if $0 < \lambda < \frac{1}{\lambda^*}$, then for all $\varepsilon > 0$ there exists an N_ε such that $n \geq N_\varepsilon$ implies

$$\frac{1}{\lambda^*} - \varepsilon \leq \mu_n \leq \frac{1}{\lambda_*} + \varepsilon \ . \qquad \text{(G.3)}$$

Let $r = \frac{\lambda^*}{\lambda_*}$. Then,

$$\frac{1}{\lambda^*} - \varepsilon \leq \mu_n \leq \frac{r}{\lambda^*} + \varepsilon \ .$$

Since $1 < r < 2$ by hypothesis, for any $0 < \varepsilon < \frac{2-r}{\lambda^*}$ there exists a $\delta > 0$, namely

$$\delta = \frac{2-r}{\lambda^*} - \varepsilon$$

such that (G.1) is satisfied for sufficiently large n . Hence,

$$\lim_{n\to\infty} Z_n = 0 \ .$$

<u>This completes the proof of the theorem.</u>

An interesting question arises at this point. By changing μ according to (F.5), is the rate of convergence of (F.6) increased over that of using a constant $\mu = \lambda$? A partial answer is given by the following argument. Note that

$$\|Z_{n+1}\|^2 = \|(I - \lambda R)Z_n\|^2 + (1 - \lambda\alpha_{n-1})^2\mu_{n-1}^2\|RZ_{n-1}\|^2$$

$$-2(1 - \lambda\alpha_{n-1})\mu_{n-1}Z_n^T R(I - \lambda R)Z_n \ .$$

Therefore, if

$$2(1 - \lambda\alpha_{n-1})\mu_{n-1}Z_n^T R(I - \lambda R)Z_n \geq (1 - \lambda\alpha_{n-1})^2\mu_{n-1}^2\|RZ_{n-1}\|^2 \ , \qquad \text{(G.4)}$$

then the rate of convergence of the sequence $\{Z_n\}$ has been speeded up by changing the step size μ. If $0 < \lambda < \frac{1}{\lambda^*}$, then one can write (G.4) as

$$2Z_n^T R Z_n \geq [(1 - \lambda\alpha_{n-1})\mu_{n-1} + 2\lambda] Z_n^T R^2 Z_n .$$

This inequality holds if for all the eigenvalues of R, λ_i, it is the case

$$2\lambda_i \geq [(1 - \lambda\alpha_{n-1})\mu_{n-1} + 2\lambda]\lambda_i^2$$

or

$$\frac{2}{\lambda_i} \geq [(1 - \lambda\alpha_{n-1})\mu_{n-1} + 2\lambda] .$$

From the proof of the theorem, if $\mu_1 \leq \frac{1}{\lambda_*}$, then $\mu_n \leq \frac{1}{\lambda_*}$ for all n. Consequently,

$$(1 - \lambda\alpha_{n-1})\mu_{n-1} + 2\lambda \leq (1 - \lambda\lambda_*)\frac{1}{\lambda_*} + 2\lambda$$

$$= \frac{1 + \lambda\lambda_*}{\lambda_*} .$$

Hence, if

$$\frac{2}{\lambda_i} \geq \frac{1 + \lambda\lambda_*}{\lambda_*}$$

the rate of convergence has been increased. Since

$$\frac{2}{\lambda_i} \geq \frac{2}{\lambda^*}$$

for all $1 \leq i \leq p$, a sufficient condition for speed-up is

$$\frac{1}{\lambda^*} \geq \frac{1 + \lambda\lambda_*}{\lambda_*}$$

or

$$\lambda \leq \frac{2\frac{\lambda_*}{\lambda^*} - 1}{\lambda_*} \quad .$$

Therefore one wants

$$\frac{\lambda^*}{\lambda_*} < 2 \ ,$$

$$0 < \mu_1 \leq \frac{1}{\lambda_*} \quad ,$$

and

$$0 < \lambda < \min\left[\frac{\frac{\lambda_*}{\lambda^*} - 1}{\lambda_*} \ , \ \frac{1}{\lambda^*}\right] \quad .$$

It would appear that these conditions are far too restrictive for the speed-up conditions. In Fig. G.1 is plotted the convergence curves obtained experimentally for the case where the ratio λ^*/λ_* is equal to $8 (\lambda^* = 1)$.

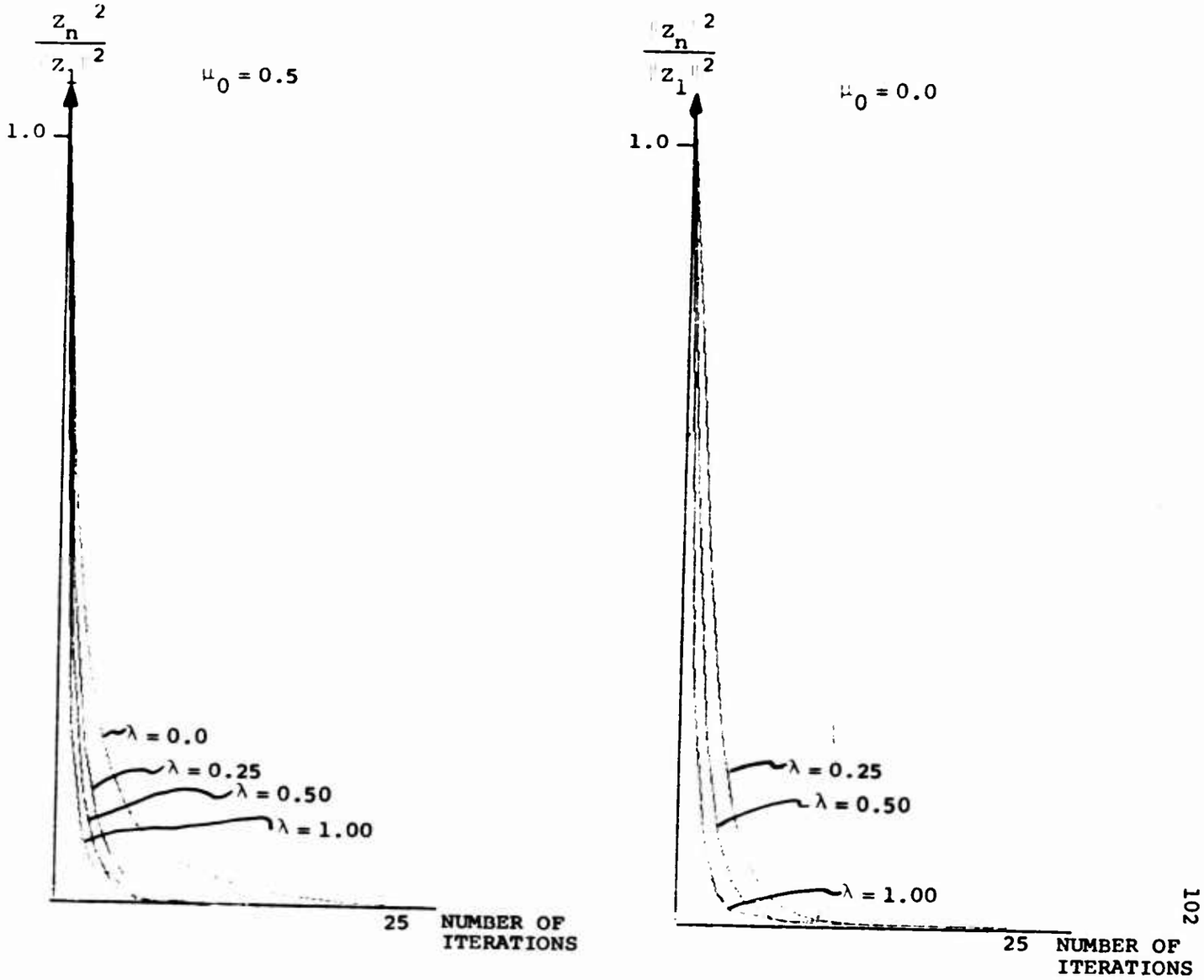


Fig. G.1. Convergence curves for algorithms (F.5) and (F.6)

APPENDIX H

AN ALTERNATE ANALYSIS OF ADAPTIVE ESTIMATION IN NONSTATIONARY ENVIRONMENTS

In the text we considered a state-space representation for the dynamic model of nonstationary statistics. Other models could also be developed. In this appendix, a first-order two-state Markov model will be considered for the single weight case. The results obtained will be shown to apply to more general scalar models.

The problem to be discussed is defined as follows. Let the sequence $\{w_n\}$ be defined by

$$w_{n+1} = w_n - \mu y_n$$

where

$$y_n = r(w_n - w_n^*) + z_n$$

$$E[z_n | w_n, w_n^*] = 0$$

$$E[z_n^2 | w_n, w_n^*] = \sigma_1^2 + \sigma_2^2 \| w_n - w_n^* \|^2$$

and

$$w_n^* = \text{least-mean-square-weight at time } n .$$

An example of this type of algorithm is the LMS adaptation algorithm discussed in Appendix E when the input data pair sequence is an independent Gaussian random process with

$$E[x_n^2 | w_n, w_n^*] = E[x_n^2] = r$$

$$E[(d_n - w_n^* x_n)^2] = \sigma_1^2 / r$$

and

$$E[(x_n^2 - r)^2 | w_n, w_n^*] = \sigma_2^2 .$$

Let the optimum weights $\{w_n^*\}$ be generated by a first-order N-state Markov process with transition probability matrix $P = \{p_{ij}\}$. If, at time n, the Markov process is in state i, then $w_n^* = \varphi_i$. The steady-state probability vector is given by

$$\pi = [\pi_1, \pi_2, \ldots \pi_N]$$

where

$$\pi_i = \text{steady-state probability that } w_n^* = \varphi_i .$$

As an example, consider the two-state Markov process shown in Fig. H.1. If at time n the Markov process is in state 1 (indicated by φ_1), then $w_n^* = \varphi_1$; if at time n, the Markov process is in state 2 (indicated by φ_2), then $w_n^* = \varphi_2$. The transition probability matrix is given by

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix} = \begin{bmatrix} 1-p & p \\ q & 1-q \end{bmatrix}$$

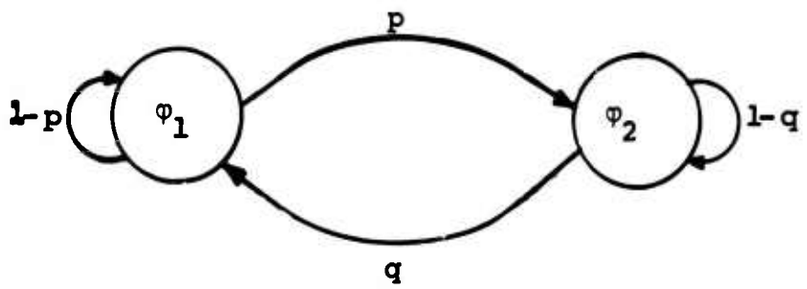
The steady-state probability vector is given by [60]

$$\pi = [\pi_1, \pi_2] = \frac{q}{p+q} \ , \ \frac{p}{p+q}$$

To demonstrate the tracking ability of the algorithm, consider first the random process $\{w_n\}$ defined by

$$w_{n+1} = w_n - \mu r (w_n - w_n^*) \ ,$$

i.e., the noiseless gradient descent with $z_n = 0$. The behavior of $b_n^2 = E[\|w_n - w_n^*\|^2]$ is summarized by



$$W_n^* = \begin{cases} \varphi_1 & \text{prob} \quad \Pi_1 = \dfrac{q}{p+q} \\ \varphi_2 & \text{prob} \quad \Pi_2 = \dfrac{p}{p+q} \end{cases}$$

Fig. H.1. Two-state Markov process.

Theorem 1. Define the random process $\{w_n\}$ by

$$w_{n+1} = w_n - \mu r (w_n - w_n^*)$$

where w_1 is arbitrary subject to $E[(w_1 - w_1^*)^2] < \infty$. $\{w_n^*\}$ is a stationary random process with finite second moment. If $0 < \mu r < 2$, then

$$\lim_{n\to\infty} b_n^2 = \lim_{n\to\infty} (\mu r)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (1-\mu r)^{i+j} E[(w_{i+2}^* - w_1^*)(w_{j+2}^* - w_1^*)] .$$

Proof of the Theorem 1.

Note that by successive iterations, the algorithm can be written

$$w_{n+1} - w_{n+1}^* = (1-\mu r)^n (w_1 - w_{n+1}^*) + \mu r \sum_{i=1}^{n} (1-\mu r)^{n-i} (w_i^* - \varphi_k) .$$

Thus, if $|1-\mu r| < 1$, then

$$b_{n+1}^2 = E[(w_{n+1} - w_{n+1}^*)^2]$$

$$= 0(n) + (\mu r)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} (1-\mu r)^{2n-i-j} E[(w_i^* - w_{n+1}^*)(w_j^* - w_{n+1}^*)]$$

where $\lim_{n\to\infty} 0(n) = 0$. By the stationarity of $\{w_n^*\}$ one has

$$E[(w_i^* - w_{n+1}^*)(w_j^* - w_{n+1}^*)] = E[(w_{n-i+2}^* - w_1^*)(w_{n-j+2}^* - w_1^*)] .$$

By re-indexing the expression for b_{n+1}^2, one obtains the result

$$b_{n+1}^2 = 0(n) + (\mu r)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{r-1} (1-\mu r)^{i+j} c(i,j) ,$$

where

$$c(i,j) = E[(w^*_{i+2} - w^*_1)(w^*_{j+2} - w^*_1)] .$$

The conclusion of the theorem follows immediately.

This completes the proof of Theorem 1.

It should be emphasized that this theorem holds for any stationary random process $\{w^*_n\}$ with finite second moment. The following corollary gives one a way of evaluating the expression when:

Corollary 1. If the $\{w^*_n\}$ are generated by an N-state, first-order, stationary Markov process and $\{w_n\}$ are as in Theorem 1, then

$$\lim_{n\to\infty} b_n^2 = (\mu r)^2 \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (1-\mu r)^{i+j} \varphi^T \left[\overbrace{I - P^{j+1} - P^{i+1} + P^{|i-j|}}\right] \varphi$$

where

$$\widetilde{P^m} = \left\{\pi_i P^{(m)}_{ij}\right\}$$

and $P^{(m)}_{ij}$ is an element of $P^m = \underbrace{P \cdot \cdot \cdot P}_{m \text{ factors}}$.

Proof of Corollary 1.

The corollary follows immediately from:

Lemma. Let $\{\theta_n\}$ be a first-order, N-state, stationary Markov process with transition matrix $P = \{P_{ij}\}$. Let the elements of $(P)^m$ be denoted by $P^{(m)}_{ij}$. Define

$$\widetilde{(P)^m} = \{\pi_i P^{(m)}_{ij}\}$$

and

$$\varphi = \begin{bmatrix} \varphi_1 \\ \cdot \\ \cdot \\ \cdot \\ \varphi_N \end{bmatrix} = \text{state vector.}$$

Then

$$E[\theta_{n+m}\theta_n] = \varphi^T \widetilde{(P)^m} \varphi \ .$$

Proof of the Lemma.

By straightforward calculation

$$E[\theta_{n+m}\theta_n] = \sum_{k=1}^{N} \sum_{\ell=1}^{N} \varphi_k \varphi_\ell P\{\theta_{n+m} = \varphi_k \ , \ \theta_n = \varphi_\ell\}$$

$$= \sum_{k=1}^{N} \sum_{\ell=1}^{N} \varphi_k \varphi_\ell \pi_\ell P_{\ell k}^{(m)}$$

$$= \varphi^T \widetilde{(P)^m} \varphi \ .$$

This completes the proof of the lemma.

Noting that the $[\widetilde{\ \cdot\ }]$ operation is linear, it follows immediately that

$$\lim_{n\to\infty} b_n^2 = \lim_{n\to\infty} (\mu r)^2 \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (1-\mu r)^{i+j} \varphi^T \widetilde{\left[I - P^{j+1} - P^{i+1} + P^{|i-j|} \right]} \varphi$$

or by symmetry

$$\lim_{n\to\infty} b_n^2 = \lim_{n\to\infty} (\mu r)^2 \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (1-\mu r)^{i+j} \varphi^T \overbrace{\left[I - 2P^{i+1} + P^{|i-j|} \right]} \varphi \ .$$

This completes the proof of Corollary 1.

The following corollary provides a nice example of the tracking ability of the algorithm.

Corollary 2. Under the assumptions of Corollary 1 for the two-state Markov process considered in Fig. 1,

$$\lim_{n\to\infty} b_n^2 = \pi_1\pi_2(\varphi_1 - \varphi_2)^2 \frac{2(p+q)}{(2-\mu r)[\mu r(1-p-q) + (p+q)]}$$

Proof of Corollary 2.

Using the result [ ], $P^m = \alpha^m \begin{bmatrix} \pi_2 & -\pi_2 \\ -\pi_1 & \pi_1 \end{bmatrix}$, one has

$$I - P^{i+1} - P^{j+1} + P^{|i-j|} = \left[1 - \alpha^{i+1} - \alpha^{i+1} + \alpha^{|i-j|}\right] \begin{bmatrix} \pi_2 & -\pi_2 \\ -\pi_1 & \pi_1 \end{bmatrix}$$

where $\alpha = 1-p-q$. Therefore, by Corollary 1,

$$\lim_{n\to\infty} b_n^2 = (\mu r)^2 \pi_1\pi_2(\varphi_1-\varphi_2)^2 \lim_{n\to\infty} \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (1-\mu r)^{i+j} (1-\alpha^{i+1}-\alpha^{j+1}+\alpha^{|i-j|}) \ .$$

The only term of any difficulty in evaluating is

$$\lim_{n\to\infty} \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} (1-\mu r)^{i+j} \alpha^{|i-j|} \ .$$

Defining $\beta = 1 - \mu r$, one can show

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \alpha^{|i-j|}\beta^{i+j} = \sum_{j=-(n-1)}^{n-1} \frac{\beta^{|j|}\alpha^{|j|}}{1-\beta^2} - \sum_{j=-(n-1)}^{n-1} \frac{\beta^{2n}\beta^{-|j|}}{1-\beta^2}\alpha^{|j|}$$

where

$$\sum_{j=-(n-1)}^{n-1} \frac{\beta^{|j|}\alpha^{|j|}}{1-\beta^2} = \frac{1}{1-\beta^2}\left\{1 + 2(\alpha\beta)\frac{1-(\alpha\beta)^{n-1}}{1-\alpha\beta}\right\}$$

and

$$\sum_{j=-(n-1)}^{n-1} \frac{\beta^{2n}\beta^{-|j|}}{1-\beta^2}\alpha^{|j|} = \begin{cases} \dfrac{\beta^{2n}}{1-\beta^2}\left\{1 + 2\beta^n(\alpha\beta^n(\alpha\beta)\dfrac{\beta^{n-1}-\alpha^{n-1}}{\beta-\alpha}\right\} & \beta\neq\alpha \\ \dfrac{\beta^{2n}}{1-\beta^2}\{2n-1\} & \beta=\alpha \end{cases}$$

In the limit as $n\to\infty$,

$$\lim_{n\to\infty} \sum_{i=0}^{n-1}\sum_{j=0}^{n-1} \alpha^{|i-j|}\beta^{i+j} = \frac{1+\alpha\beta}{1-\alpha\beta}\cdot\frac{1}{1-\beta^2} .$$

Thus, it follows after some algebra that

$$\lim_{n\to\infty} b_n^2 = \pi_1\pi_2(\Phi_1-\Phi_2)^2 \frac{2(p+q)}{(2-\mu r)(\mu r(1-p-q)+(p+q)}\cdot$$

This completes the proof of the corollary.

The expected squared distance between the optimum constant weight and the $\{w_n^*\}$ is given by

Corollary 3. Let $\{w_n^*\}$ be as in Corollary 2. Let w_0^* be that constant weight which minimizes $E[(w_n^*-w_0^*)^2]$. Then

$$w_0^* = \pi_1 \varphi_1 + \pi_2 \varphi_2$$

and

$$E[(w_n^* - w_0^*)^2] = \pi \ .$$

Proof of Corollary 3.

It is a well-known result [61] that the constant which minimizes $E[(x-a)^2]$ is

$$a = E[x] \ .$$

Therefore,

$$w_0^* = \pi_1 \varphi_1 + \pi_2 \varphi_2$$

and

$$E[(w_n^* - w_0^*)^2] = \pi_1 (\varphi_1 - \pi_1 \varphi_1 - \pi_2 \varphi_2)^2 + \pi_2 (\varphi_2 - \pi_1 \varphi_1 - \pi_2 \varphi_2)^2$$

$$= \pi_1 \pi_2 (\varphi_1 - \varphi_2)^2 \ .$$

This completes the proof of Corollary 3.

Define

$$P(\mu) = \lim_{n \to \infty} b_n^2 - E[(w_n^* - w_0^*)^2] \ .$$

In Fig. H.2 is plotted $\dfrac{P(\mu)}{E[(w_n^* - w_0^*)^2]}$ for three values of $p+q$. It should be noted that when this expression is negative, the $\{w_n\}$ has smaller misadjustment than is possible with any fixed weight vector.

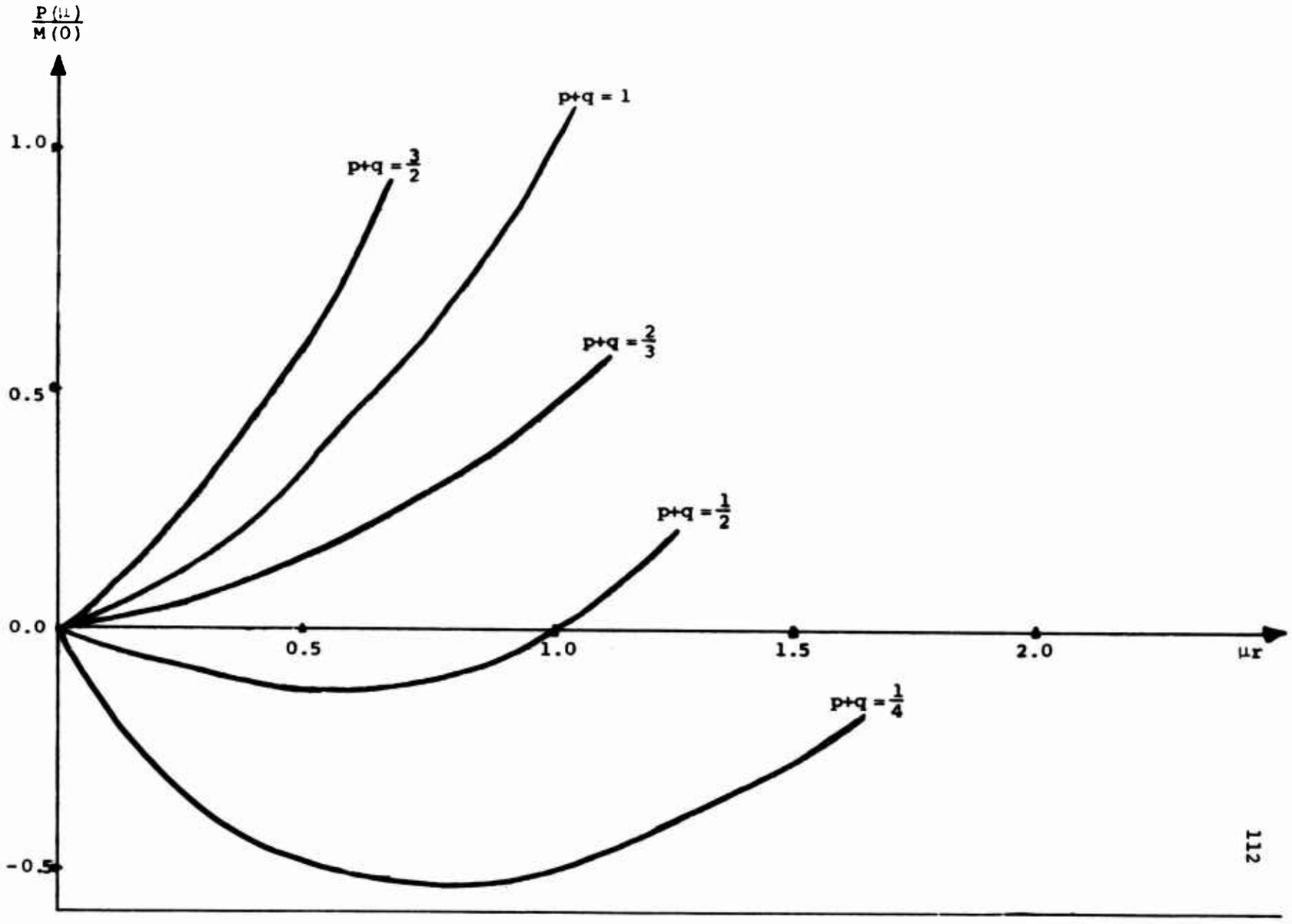


Fig.H.2. Plot of normalized misadjustment as function of μr .

Adding noise to the gradient estimate, as originally discussed, one has

Theorem 2. Define the random process $\{w_n\}$ by

$$w_{n+1} = w_n - \mu y_n$$

where

$$y_n = r(w_n - w_n^*) + z_n$$

where z_n satisfies the three conditions

$$E[z_n | w_n, w_n^*] = 0$$

$$E[z_n^2 | w_n, w_n^*] = \sigma_1^2 + \sigma_2^2 (w_n - w_n^*)^2$$

$$E[z_i z_j] = 0 \quad i \neq j \quad .$$

Let the random process $\{w_n^*\}$ be stationary with finite second moment. If $0 < \mu < \frac{2r}{r^2 + \sigma_2^2}$, then

$$\lim_{n\to\infty} b_n^2 = \frac{r(2-\mu r)}{2r - \mu(r^2+\sigma_2^2)} (\mu r)^2 \lim_{n\to\infty} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (1-\mu r)^{i+j} C(i,j)$$

$$+ \frac{\mu\sigma_1^2}{2r - \mu(r^2+\sigma_2^2)} \quad .$$

Proof of Theorem 2.

Proceeding as in the proof of Theorem 1,

$$w_{n+1} - w_{n+1}^* = (1-\mu r)^n (w_1 - w_{n+1}^*) + \mu \sum_{i=1}^{n} (1-\mu r)^{n-i} [r(w_i^* - w_{n+1}^*) - z_i] \quad .$$

Squaring and taking the expectation yields

$$b_{n+1}^2 = 0(n) + (\mu r)^2 \sum_{i=1}^{n} \sum_{j=1}^{n} (1-\mu r)^{2n-i-j} E[(w_i^* - w_{n+1}^*)(w_j^* - w_{n+1}^*)]$$

$$+ \mu^2 \sum_{i=1}^{n} (1-\mu r)^{2(n-i)} [\sigma_1^2 + \sigma_2^2 b_i^2] \ .$$

From this expression a lower and an upper bound on the $\{b_n^2\}$ may be obtained. After a little algebra it can be shown that both bounds are equal and given by

$$b^2 = \frac{r(2-\mu r)(\mu r)^2}{2r - \mu(r^2+\sigma_2^2)} \lim_{n\to\infty} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (1-\mu r)^{i+j} C(i,j)$$

$$+ \frac{\mu\sigma_1^2}{2r - \mu(r^2+\sigma_2^2)} \ ,$$

where

$$C(i,j) = E[(w_{i+2}^* - w_1^*)(w_{j+2}^* - w_1^*)] \ .$$

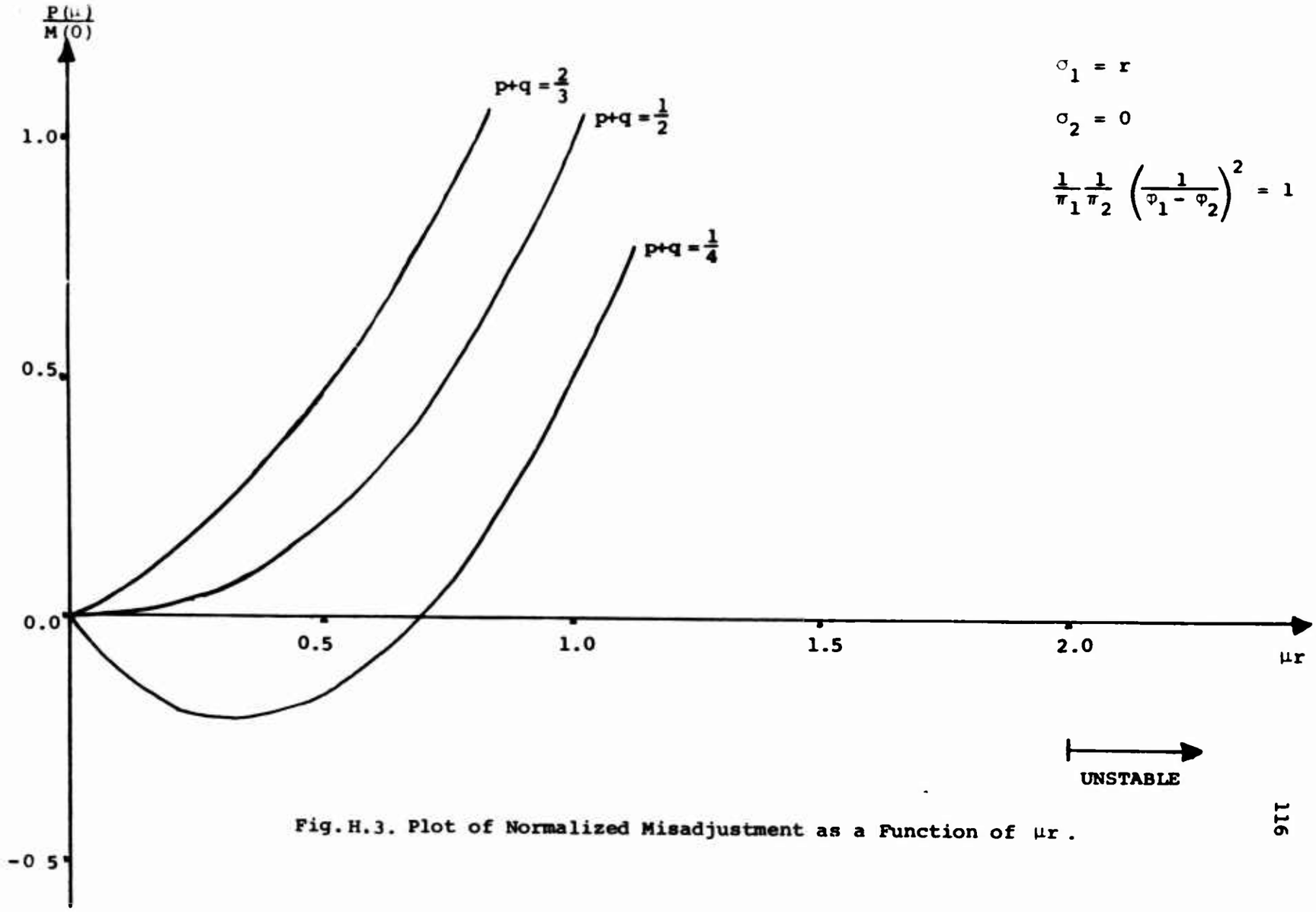
This completes the proof of Theorem 2.

For the two-state Markov process previously considered we have plotted

$$\frac{P(\mu)}{E[(w_n^* - w_0^*)^2]}$$

for the noisy gradient case with $\sigma_2 = 0$. The effect of the gradient noise, z_n , is to increase the misadjustment and decrease the magnitude of the optimum μ to use for a given

(p,q) pair. This is strikingly evident by comparing the graphs of the noiseless gradient descent procedure (Fig. H.2) with the noisy gradient descent procedure (Fig. H.3)



$\sigma_1 = r$

$\sigma_2 = 0$

$\frac{1}{\pi_1}\frac{1}{\pi_2}\left(\frac{1}{\varphi_1 - \varphi_2}\right)^2 = 1$

Fig. H.3. Plot of Normalized Misadjustment as a Function of μr .

APPENDIX I

THE ADAPTATION ALGORITHM AS A FILTER

This appendix is concerned with looking at the time-varying filtering problem from an entirely different viewpoint than that of the text. It will be shown how the adaptation algorithm (3.5) can be used as a filter. The performance of this system will be compared to the optimum Kalman system for a scalar filtering problem.

A. PROBLEM STATEMENT AND ASSUMPTIONS

It will be assumed that the target signal and noise field can be modeled by the dynamic systems

$$\theta_S(n+1) = F_S\,\theta_S(n) + G_S\,U_S(n) \qquad \text{(I.1a)}$$

$$S(n) = H_S\,\theta_S(n) + V_S(n) \qquad \text{(I.1b)}$$

and

$$\theta_N(n+1) = F_N\,\theta_N(n) + G_N\,U_N(n) \qquad \text{(I.2a)}$$

$$N(n) = H_N\,\theta_N(n) + V_N(n) \qquad \text{(I.2b)}$$

where θ_S and θ_N are the state-vectors, F_S and F_N are known matrices, U_S and U_N are random vector inputs of zero mean satisfying $E[U_S(n)U_S^T(m)] = Q_S\delta_{nm}$, $E[U_N(n)U_N^T(m)] = Q_N\delta_{nm}$, $E[U_N(n)U_S^T(m)] = 0$, G_S and G_N are known shaping matrices, H_S and H_N are known output matrices, V_S and V_N are random noise vectors of zero mean satisfying $E[V_S(n)V_S^T(m)] = R_S\delta_{nm}$, $E[V_N(n)V_N^T(m)] = R_N\delta_{nm}$, $E[V_S(n)V_N(m)] = 0$, and $S(n)$ and $N(n)$ are the signal

and noise components present on the array sensors at time n. The received vector at time n is $X(n) = S(n) + N(n)$ (see Fig. I.1). Note the change in notation from that presented in the test. It will aslo be assumed that V_S, V_N, U_S, and U_N are mutually independent Gaussian random vectors. This is the usual Kalman model for dynamic linear discrete-time random processes.

For the combined system model, (I.1a), (I.1b), (I.2a), and (I.2b), one has

$$\theta(n+1) = F\theta(n) + GU(n) \qquad \text{(I.3a)}$$

$$X(n) = H\theta(n) + V(n) \qquad \text{(I.3b)}$$

where

$$\theta(n) = \begin{bmatrix} \theta_S(n) \\ \theta_N(n) \end{bmatrix}$$

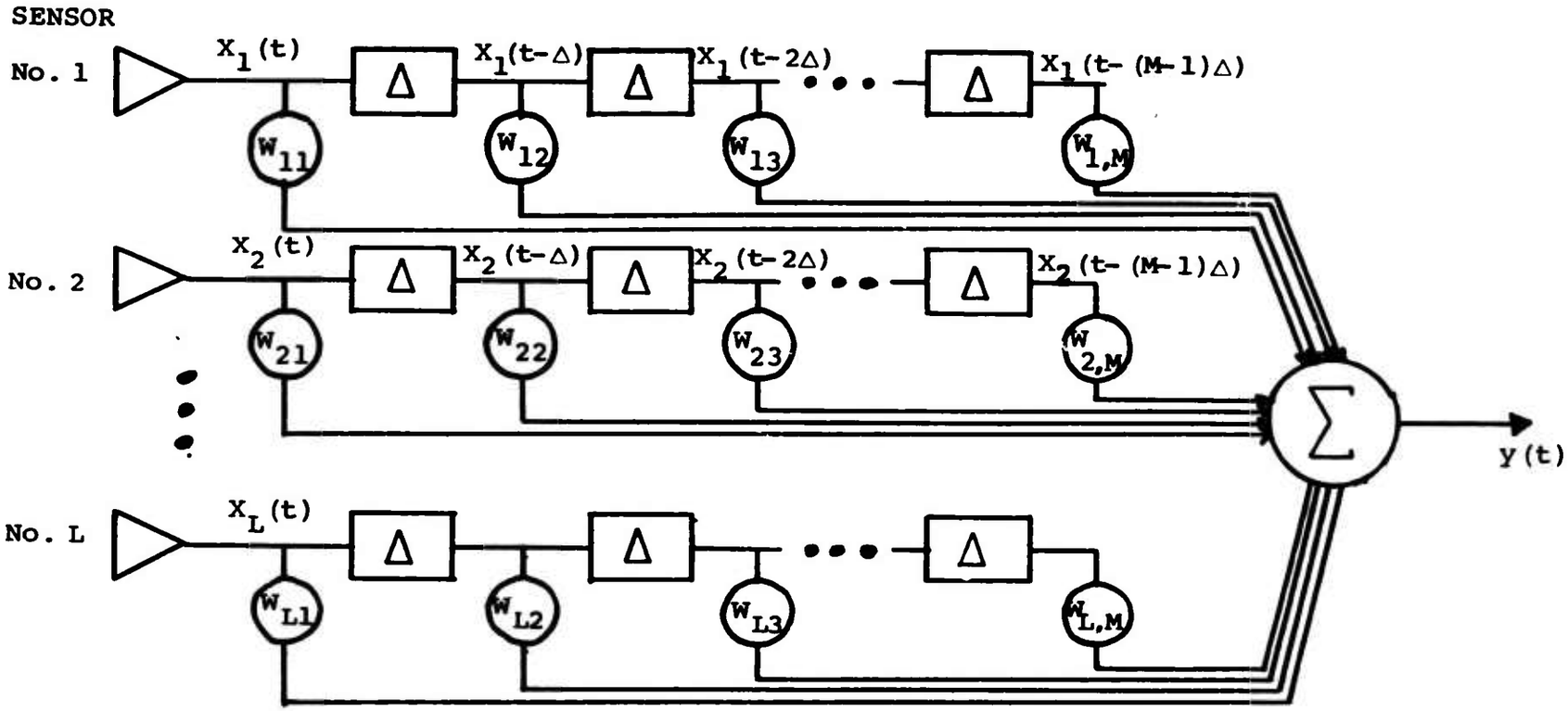
$$F = \begin{bmatrix} F_S & 0 \\ 0 & F_N \end{bmatrix}$$

$$G = \begin{bmatrix} G_S & 0 \\ 0 & G_N \end{bmatrix}$$

$$U(n) = \begin{bmatrix} U_S(n) \\ U_N(n) \end{bmatrix}$$

$$H = \begin{bmatrix} H_S & H_N \end{bmatrix}$$

$$V(n) = V_S(n) + V_N(n)$$



$$X(t) \triangleq \begin{bmatrix} X_1(t) \\ X_2(t) \\ \vdots \\ X_L(t) \end{bmatrix} \qquad W_i \triangleq \begin{bmatrix} W_{1i} \\ W_{2i} \\ \vdots \\ W_{Li} \end{bmatrix}$$

Fig. I.1. The array problem.

This system, (I.3a) and (I.3b), is shown in Fig. I.2.

The problem is to determine the filter $\{W(n+1,j):j=1,2,..n\}$ which minimizes

$$E[\|S(n+1) - \hat{S}(n+1|n)\|^2] \tag{I.4}$$

where $\hat{S}(n+1|n)$ is the estimate of $S(n+1)$ given by

$$\hat{S}(n+1|n) = \sum_{j=1}^{n} W(n+1,j)X(j) \ .$$

The $\hat{S}(n+1|n)$ minimizing (I.4) is called the <u>minimum-variance estimator</u> of $S(n+1)$.

B. <u>THE OPTIMUM RECURSIVE ESTIMATION FILTER</u>

As shown in [57], the minimum-variance estimator of $S(n+1)$, denoted from here on by $\hat{S}(n+1|n)$, is given by

$$\hat{S}(n+1|n) = \tilde{H}\hat{\theta}(n+1|n) \tag{I.5}$$

where

$$\tilde{H} = [H_s \ 0] \tag{I.6}$$

and $\hat{\theta}(n+1|n)$ is the minimum-variance estimator of $\theta(n+1)$ given the observations $X(1),X(2),...,X(n)$.

The optimal Kalman recursive linear filter for estimating $\theta(n)$ based on the observations $X(1),X(2),...X(n-1)$ is given by [55] - [57]

$$\hat{\theta}(n+1|n) = F\left[P(n)H^T\left(HP(n)H^T + R\right)^{-1}\left(X(n) - H\hat{\theta}(n|n-1)\right) + \hat{\theta}(n|n-1)\right] \tag{I.7a}$$

$$P(n+1) = F\left[P(n) - P(n)H^T(HP(n)H^T + R)^{-1} HP(n)\right]F^T + GQG^T \tag{I.7b}$$

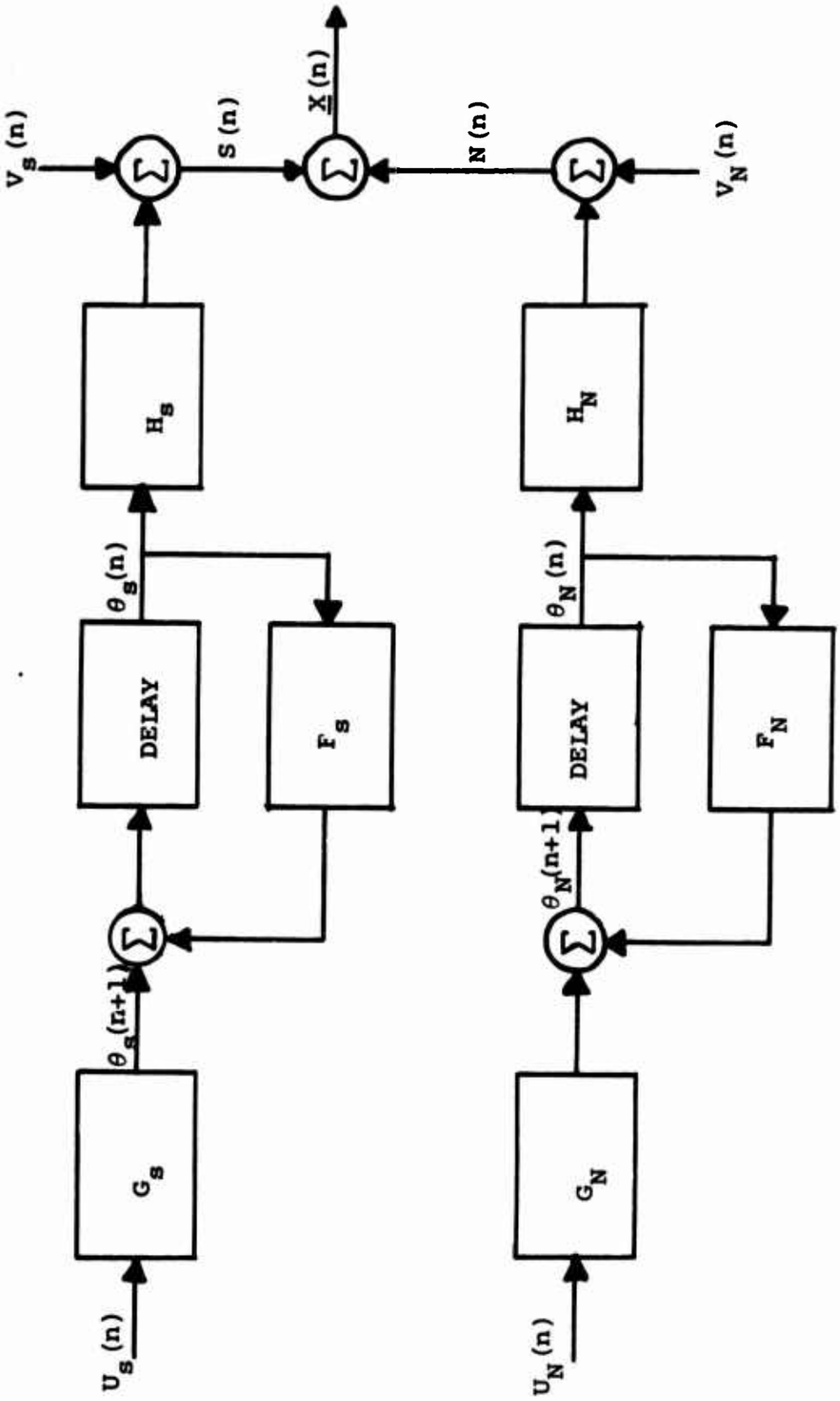


Fig. I.2. Data model for array problem

where $\hat{\theta}(n|n-1)$ is the minimum-variance estimate of $\theta(n)$ given the data observations $X(1),X(2),\ldots X(n-1)$. $P(n)$ is the error covariance matrix at time n defined by

$$P(n) = E\left[\left(\theta(n) - \hat{\theta}(n|n-1)\right)\left(\theta(n) - \hat{\theta}(n|n-1)\right)^T\right] .$$

An equivalent form for the Kalman filter given by (1.7a) and (1.7b) is

$$\hat{\theta}(n+1|n) = F\left[\hat{\theta}(n|n-1) - K(n)\left(H\hat{\theta}(n|n-1) - X(n)\right)\right] \qquad (I.8a)$$

$$P(n+1) = F[I - K(n)H]P(n)F^T + GQG^T \qquad (I.8b)$$

$$K(n) = P(n)H^T(HP(n)H^T + R)^{-1} . \qquad (I.8c)$$

(See Fig. I.3).

C. A RECURSIVE FEEDBACK FILTER BASED ON THE ADAPTATION ALGORITHM

Consider now a suboptimum approach for estimating $S(n)$ based on the observations $X(1),X(2),\ldots,X(n-1)$. The idea is to first estimate $\theta_s(n)$ by $\hat{\theta}_s(n)$, say. The estimate of $S(n)$ will be defined by

$$\hat{S}(n) = H_s\hat{\theta}_s(n) . \qquad (I.9)$$

In analogy to (2.3), define the mean-squared error at time n by

$$\xi_n \triangleq E[\|S(n) - \hat{S}(n)\|^2] . \qquad (I.10)$$

Using (I.1b) and (I.9) in (I.10), one has the expression

$$\xi_n = E[\|H_s[\theta_s(n) - \hat{\theta}_s(n)] + V_s(n)\|^2] \qquad (I.11)$$

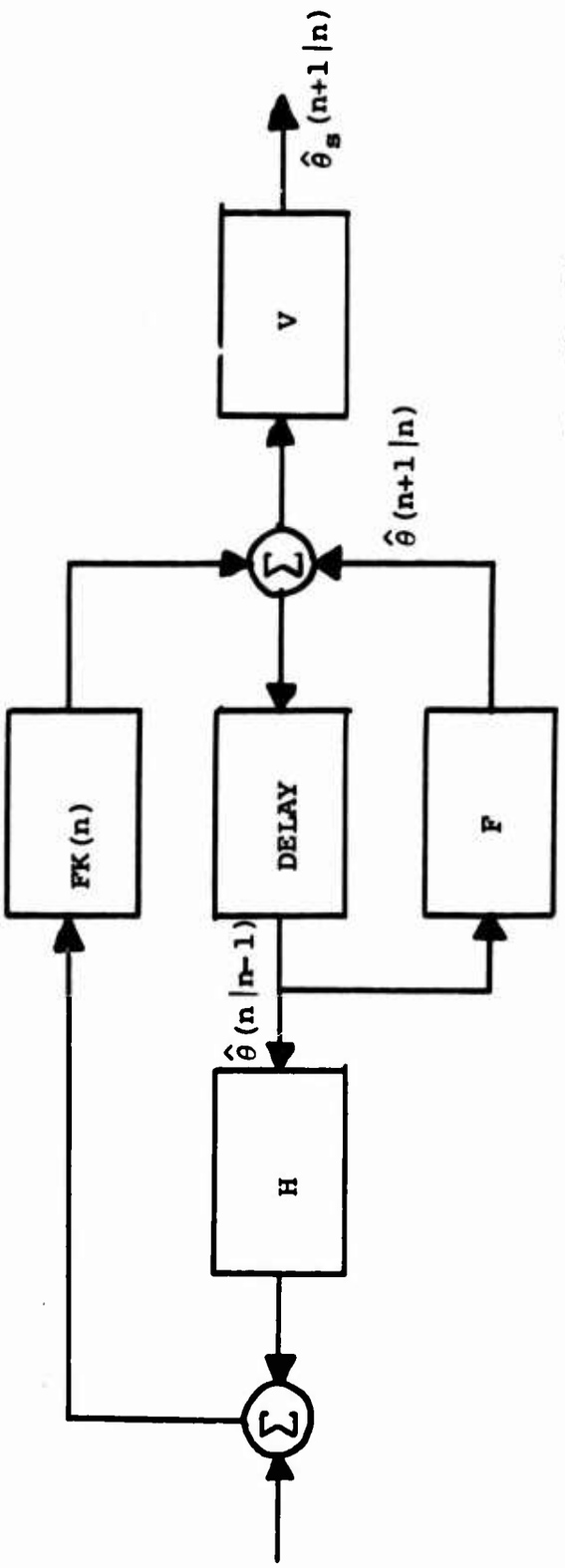


Fig. I.3. Optimum minimum-variance array processor.

An unbiased estimate of the gradient of (I.11) with respect to $\hat{\theta}_s(n)$ is given by

$$Y_s(n) = H_s^T H_s[\hat{\theta}_s(n) - \hat{\theta}_s(n)] - H_s^T V_s(n)$$

$$= H_s^T[\hat{S}(n) - S(n)]. \qquad (I.12)$$

The corresponding algorithm for estimating $\theta_s(n+1)$, based on the method of steepest descent, is

$$\hat{\theta}_s(n+1) = F_s[\hat{\theta}_s(n) - \mu_n Y_s(n)] . \qquad (I.13)$$

This algorithm is of the form (3.5) in the text.

Note that the algorithm (I.13) requires knowledge of either $\theta_s(n)$ or $S(n)$ in addition to the signal model parameters. If the object were to estimate $\theta_s(n)$ based on $S(1),\ldots S(n-1)$, this restriction would be acceptable. However, in general, one does not know the signal sequence $\{S(k)\}$.

Suppose we estimate the total state vector $\theta(n)$ by $\hat{\theta}_n$. The estimate of $S(n)$ will be defined by

$$\hat{S}(n) = \tilde{H}\hat{\theta}_n \qquad (I.14)$$

where $\tilde{H}$ is given by (I.6). Define the mean-squared error at time n by

$$\xi_n \triangleq E[\|X(n) - \hat{X}(n)\|^2]$$

where

$$\hat{X}(n) = H\hat{\theta}_n \ .$$

Proceeding as above, the algorithm for estimating $\theta(n+1)$ is

$$\hat{\theta}_{n+1} = F[\hat{\theta}_n - \mu_n Y_n] \tag{I.15}$$

where

$$Y_n = H^T H[\hat{\theta}_n - \theta_n] - H^T V_n$$

$$= H^T H\hat{\theta}_n - H^T X_n \ . \tag{I.16}$$

Comparing (I.15) with the Kalman filter (I.8a), one should note that they are equivalent if

$$K(n) = \mu_n H^T \ .$$

(See Figs. I.3 and I.4.)

The performance of the filter given by (I.15) and (I.16) is summarized by:

<u>Theorem 1.</u> Let $\{\hat{\theta}_n\}$ be as defined by (I.15) and (I.16) with $\mu_n = \mu$. If

$$\lambda^2_{max}[F(I - \mu HH^T)] < 1 \ ^{\dagger}$$

then

$$\limsup_{n\to\infty} E[\|\hat{\theta}_n - \theta_n\|^2] \leq \frac{\mu^2 tr[FH^T RHF^T] + tr[GQG^T]}{1 - \lambda^2_{max}[F(I - \mu H^T H)]} \tag{I.17}$$

$^{\dagger}\lambda^2_{max}(A)$ is defined to be the maximum eigenvalue of the matrix $A^T A$.

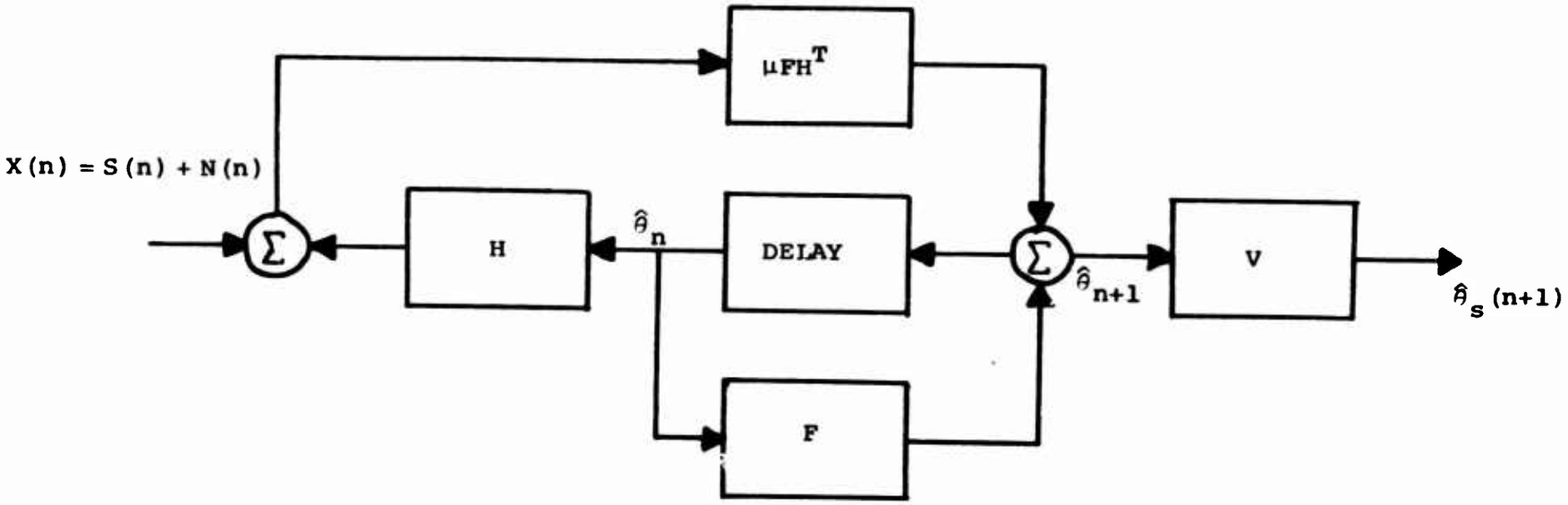


Fig. I.4. An adaptive feedback filter based on the LMS algorithm.

Proof of the Theorem.

Subtracting θ_{n+1} from both sides of (I.15) one obtains

$$\hat{\theta}_{n+1} - \theta_{n+1} = F[I - \mu H^T H](\hat{\theta}_n - \theta_n) + \mu F H^T V_n - G U_n \quad . \qquad (I.18)$$

Defining

$$B_n = E[(\hat{\theta}_n - \theta_n)(\hat{\theta}_n - \theta_n)^T]$$

it can be easily shown from (I.17) that

$$B_{n+1} = F[I - \mu H^T H] B_n [I - \mu H^T H] F^T + \mu^2 F H^T R H F^T + G Q G^T \quad .$$

Hence,

$$tr\{B_{n+1}\} = tr\{A^T A B_n\} + \mu^2 tr\{F H^T R H F^T\} + tr\{G Q G^T\} \qquad (I.19)$$

where

$$A = F[I - \mu H^T H] \quad .$$

If $\|A^T A\| < 1$, or $\lambda^2_{max}(A) < 1$, then

$$tr\{B_{n+1}\} \leq \lambda^2_{max}(A) tr\{B_n\} + \mu^2 tr\{F H^T R H F^T\} + tr\{G Q G^T\} \quad .$$

Hence, it follows

$$\limsup_{n\to\infty} tr\{B_n\} \leq \frac{\mu^2 tr\{F H^T R H F^T\} + tr\{G Q G^T\}}{1 - \lambda^2_{max}(A)} \qquad (I.17)$$

This completes the proof of the theorem.

Remark: A tighter bound can be obtained starting from

$$B_{n+1} = AB_nA^T + S$$

where

$$S = \mu^2 FH^T RHF^T + GQG^T .$$

It can be shown that if $\lambda^2_{max}(A) < 1$, then $\{B_n\}$ converges to some matrix, say B .† Hence,

$$B = ABA^T + S .$$

Taking the trace, we conclude

$$tr\{B\} = tr\{ABA^T\} + tr\{S\}$$

$$= tr\{A^TAB\} + tr\{S\}$$

and consequently

$$tr\{B\} \leq \frac{tr\{S\}}{\lambda_{min}(I - A^TA)} . \qquad (I.20)$$

Remark: The corresponding result for the Kalman filter is

$$P_\infty = F[P_\infty - P_\infty H^T(HP_\infty H^T + R)^{-1}HP_\infty]F^T + GQG^T . \qquad (I.21)$$

In general, no way has yet been bound to compare these two results. However, the following scalar example does provide an interesting comparison.

---

†This result is a direct consequence of fixed-point theory [58] as applied to the vector space of matrices with the norm on this space defined by $\sqrt{tr(A^TA)}$.

D. COMPARISON OF KALMAN AND SUBOPTIMUM FILTERS

Let θ_n be the signal at time n generated according to

$$\theta_{n+1} = \gamma\theta_n + U_n$$

and let x_n be the received signal at time n given by

$$x_n = \theta_n + V_n .$$

By (I.8a), (I.8b), and (I.8c), the Kalman estimates are given by

$$\hat{\theta}_{n+1} = \gamma\left[\hat{\theta}_n - \left(\frac{P_n}{P_n + r}\right)\left(\hat{\theta}_n - x_n\right)\right]$$

$$P_{n+1} = \gamma^2 \frac{P_n r}{P_n + r} + q$$

and by (I.15) and (I.16), the suboptimum estimates

$$\hat{\theta}_{n+1} = \gamma(\hat{\theta}_n - \mu(\hat{\theta}_n - x_n))$$

$$b_{n+1} = \gamma^2(1-\mu)^2 b_n + \mu^2\gamma^2 r + q .$$

The resulting steady-state solutions are

$$P_\infty = \gamma^2 \frac{P_\infty r}{P_\infty + r} + q$$

$$b_\infty = \frac{\mu^2\gamma^2 r + q}{1 - \gamma^2(1-\mu)^2} \qquad \gamma^2(1-\mu)^2 < 1$$

Using the μ which minimizes the expression for b

$$b_\infty(\mu_{opt}) = P_\infty .$$

E. CHOOSING THE μ_n FOR SUBOPTIMUM FILTER

The convergence rate of the filter (I.15) can be improved by using a sequence $\{\mu_n\}$ rather than a constant gain μ. In fact, as pointed out earlier, if $K(n) = \mu_n H^T$ then the Kalman and LMS filters are equivalent. Since this in general will not be the case, how does one pick a good sequence $\{\mu_n\}$?

An answer to this question is found by referring back to (I.19) and minimizing the R-H-S with respect to μ. (This procedure is an extension of that given by Chein and Fu [23].) This yields

$$\mu_n = \frac{\mathrm{tr}\{F^T F H^T H B_n\}}{\mathrm{tr}\{F^T F H^T (H B_n H^T + R) H\}} \tag{I.22}$$

with the resulting recursive relation

$$\mathrm{tr}\{B_{n+1}\} = \mathrm{tr}\{F^T F B_n\} - \frac{(\mathrm{tr}\{F^T F H^T H B_n\})^2}{\mathrm{tr}\{F^T F H^T (H B_n H^T + R) H\}} + \mathrm{tr}\{G Q G^T\} .$$

For the scalar problem considered in the previous example, the μ_n found by using (I.22) are given by

$$\mu_n = \frac{b_n}{b_n + r}$$

and the optimum $K(n)$ are

$$K(n) = \frac{P_n}{P_n + r} .$$

Thus, if $b_1 = p_1$, then $\mu_n = K(n)$ for this scalar problem. It is also interesting to note that the choice of μ_n doesn't depend on knowledge of G or Q.

F. FURTHER COMPARISONS OF THE TWO FILTERS

Some further comparisons can be obtained if we assume $Q = 0$. For then the Kalman result is

$$P_\infty = F[P_\infty - P_\infty H^T (HP_\infty H^T + R)^{-1} HP_\infty]F^T$$

and consequently

$$P_\infty = 0 \ .$$

The adaptive feedback filter yields

$$\lim_{\mu \to 0} \lim_{n \to \infty} \sup E[\|\hat{\theta} - \theta_n\|^2] = 0$$

provided $\lambda^2_{max}(F) \leq 1$. Thus, in the limit both filters can be made to perform arbitrary close to each other. The previous example with $q = 0$ and $\gamma = 1$ provides a nice comparison.

Example.

Let $q = 0$ and $\gamma = 1$ in the previous example. Assume $p_1 = b_1$. The corresponding recursive equations are

$$P_{n+1} = \frac{P_n r}{P_n + r}$$

$$b_{n+1} = (1 - \mu)^2 b_n + \mu^2 r \ .$$

As shown in Appendix J,

$$P_{n+1} = \frac{P_1 r}{nP_1 + r}$$

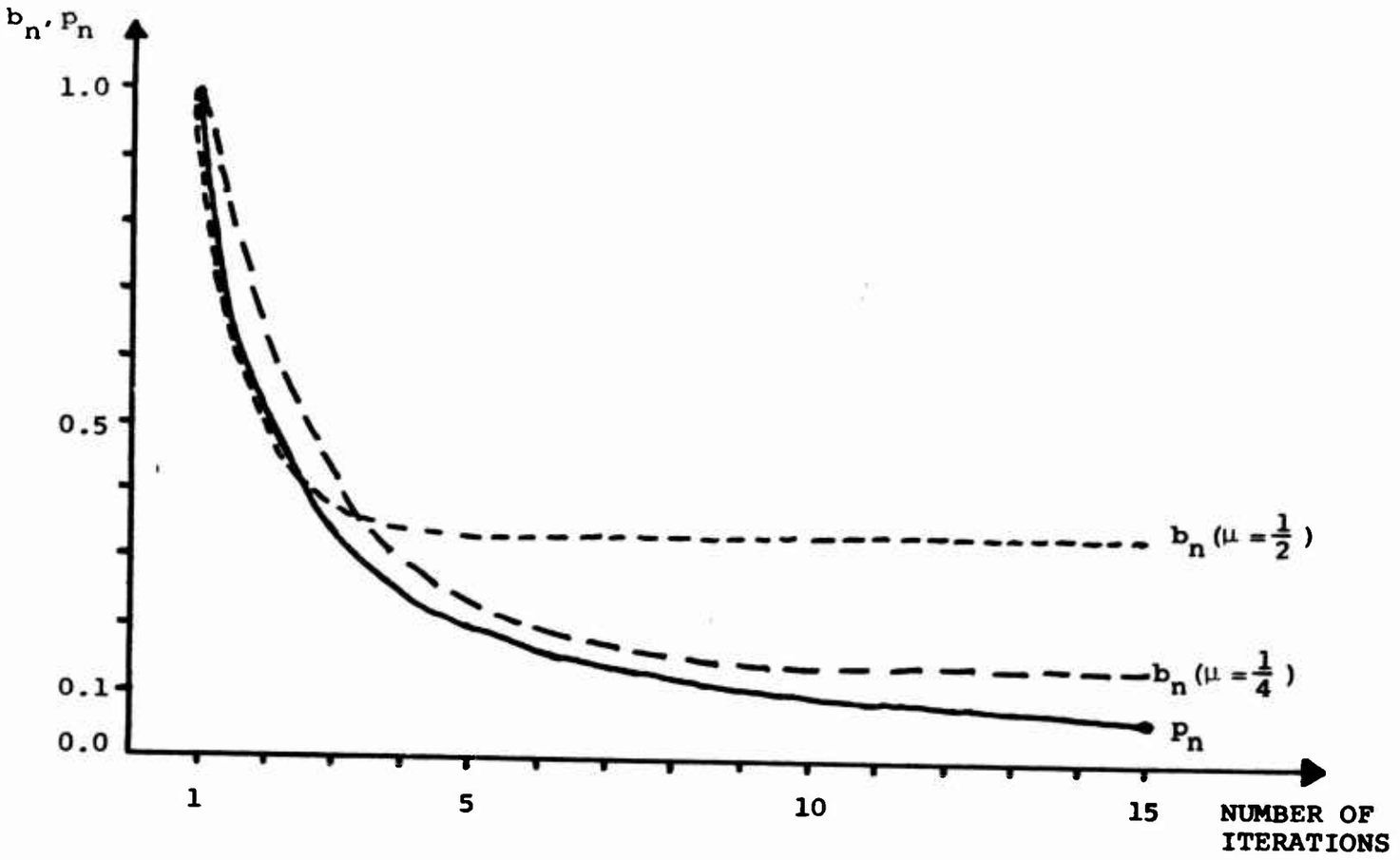


Fig. I.5. A comparison between the adaptive and Kalman filters.

and

$$b_{n+1} = (1-\mu)^{2n}\left(b_1 - \frac{\mu r}{2-\mu}\right) + \frac{\mu r}{2-\mu} \quad .$$

In Fig. I.5 we have compared P_n and b_n for two values of μ when $P_1 = b_1 = r = 1$.

Future research into the comparison of the two filters would be desirable. Although the adaptive algorithm is suboptimal, it has the advantage of being computationally simpler. Also, less a priori statistics are needed to apply this filter, the Gaussian assumption is not necessary, and the theory presented can easily be extended to a non-linear dynamic system, to name only a few advantages of the adaptive algorithm.

APPENDIX J

TWO RECURSIVE RELATIONS

The purpose of this appendix is to derive two recursive relations needed in Appendix I. They are summarized by the following lemma.

Lemma. Let $\{a_n\}$ be a sequence of non-negative real numbers. Let α and β be two positive real constants with $\alpha < 1$. Then,

(i) if $a_n = \dfrac{\beta a_n}{a_n + \beta} \qquad n \geq 1$

then $a_{n+1} = \dfrac{a_1 \beta}{n a_1 + \beta}$; and

(ii) if $a_n = \alpha a_n + \beta \qquad n \geq 1$

then $a_{n+1} = \alpha^n a_1 + \beta \dfrac{1 - \alpha^n}{1 - \alpha}$.

Proof of the Lemma.

(i) Assume that

$$a_n = \frac{a_1 \beta}{(n-1) a_1 + \beta} \quad .$$

Then

$$a_{n+1} = \frac{\beta a_n}{a_n + \beta}$$

$$= \frac{\beta^2 a_1}{(n-1)a_1 + \beta} \cdot \frac{1}{\frac{a_1\beta}{(n-1)a_1 + \beta} + \beta}$$

$$= \frac{\beta^2 a_1}{(n-1)a_1 + \beta} \cdot \frac{(n-1)a_1 + \beta}{a_1\beta + \beta(n-1)a_1 + \beta^2}$$

$$= \frac{\beta a_1}{na_1 + \beta} .$$

This is the assumed form of the relation. It is easily verified that a_1 and a_2 satisfy the formula. Hence, by induction, the desired result follows for all n.

(ii) Assume

$$a_n = \alpha^{n-1} a_1 + \beta \frac{1-\alpha^{n-1}}{1-\alpha} .$$

Then

$$a_{n+1} = \alpha a_n + \beta$$

$$= \alpha \left[\alpha^{n-1} a_1 + \beta \frac{1-\alpha^{n-1}}{1-\alpha} \right]$$

$$= \alpha^n a_1 + \beta \frac{1-\alpha^n}{1-\alpha} .$$

This is the assumed form of the relation. It is easily verified that a_1 and a_2 satisfy the formula. Hence, by

induction, the desired result follows for all n.

This completes the proof of the Lemma.

REFERENCES

1. J. P. Burg, "Three-dimensional filtering with an array of seismometers," Geophysics, 29, 5, 1964, pp. 693-713.

2. J. F. Claerbout, "Detection of P waves from weak sources a at great distances," Geophysics, 29, 2, 1964, pp. 127-211.

3. R. A. Wiggins and E. A. Robinson, "Recursive solution to the multichannel filtering problem," J. Geophys. Res., 70, 8, Apr 1965, pp. 1885-1891.

4. N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series, MIT Press, Cambridge, Mass., and John Wiley & Sons, Inc., New York, 1954.

5. F. Bryn, "Optimum signal processing of three-dimensional arrays operating on Gaussian signals and noise," J. Acoust. Soc. Am., 34, Mar 1962, pp. 289-297.

6. D. J. Edelblute, J. M. Fish, and J. L. Kinnison, "Criteria for optimum-signal-detection theory for arrays," J. Acoust. Soc. Am., 41, Jan 1967, pp. 199-205.

7. E. J. Kelly, Jr., "A comparison of seismic array processing schemes," Tech. Note DDC AD-618017 1965-21, MIT Lincoln Lab, Lexington, Mass., Jun 14, 1965.

8. H. J. Kelley, "Method of gradients," in Optimization Techniques, G. Leitmann (ed.), Academic Press, New York, pp. 206-252.

9. D. J. Wilde, Optimum Seeking Methods, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.

10. J. B. Rosen, "The gradient projection method for nonlinear programming: Linear constraints (Part I), J. Soc. Indust. Appl. Math., 8, 1, Mar 1960.

11. E. M. Glaser, "Signal detection by adaptive filters," IRE Trans., 7, Apr 1961, pp. 87-98.

12. H. J. Scudder, "Adaptive communication receivers," IEEE Trans. Information Theory, 11, Apr 1965, pp. 167-179.

13. H. L. Groginsky, L. R. Wilson, and D. Middleton, "Adaptive detection of statistical signals in noise," IEEE Trans. Information Theory, 12, Jul 1966, pp. 337-349.

14. A. V. Balakrishnan, "Effects of linear and non-linear signal processing on signal statistics," J. Res. NBS, 68D, Sep 1964, pp. 953-965.

15. L. D. Davisson, "A theory of adaptive filtering," IEEE Trans. Information Theory, 12, Apr 1966, pp. 97-102.

16. C. S. Weaver, "Adaptive communication filtering," IEEE Trans. Information Theory, 8, Sep 1962, pp. 5169-5178.

17. H. Robbins and S. Monro, "A stochastic approximation method," Ann. Math. Stat., 22, 1951, pp. 400-407.

18. J. R. Blum, "Multidimensional stochastic approximation methods," Ann. Math. Stat, 25, 1954, pp. 737-744.

19. G. E. P. Box and G. M. Jenkins, "Some statistical aspects of adaptive optimization and control," J. R. Stat. Soc., 24, Series B, 1962, pp. 297-343.

20. D. J. Sakrison, "Application of stochastic approximation methods to optimum filter design," IRE Int. Conv. Rec., 9, part 4, 1961.

21. L. A. Gardner, Jr., "Adaptive predictors," Trans. 3rd Prague Conf. on Information Theory, Stat. Dec. Functions, and Random Processes, House of Czechoslovak Academy of Sciences, Prague, 1960, pp. 123-192.

22. V. Dupac, "A dynamic stochastic approximation method," Ann. Math. Stat., 36, 1965, pp. 1695-1702.

23. Y. T. Chien, K. S. Fu, "Learning in nonstationary environment using dynamic stochastic approximation," Proc. 5th Allerton Conf. on Circuit and Systems Theory, 1967, pp. 337-345.

24. R. deFigueiredo, "Convergent algorithms for pattern recognition in non-linearly evolving nonstationary environment," Proc. IEEE (letters), 56, Feb 1968, pp. 188-9.

25. Y. C. Ho, "On stochastic approximation and optimum filtering methods," J. Math. Anal. Appl., 6, 1, Feb 1963, pp. 152-154.

26. Ya. Z. Tsypkin, "Use of the stochastic approximation method in estimating unknown distribution densities from observations," Automation and Remote Control, 26, Mar 1966, pp. 432-434.

27. Y. T. Chien, K. S. Fu, "On Bayesian learning and stochastic approximation," IEEE Trans. Syst. Sci. and Cybernetics, SCC-3, Jun 1967, pp. 23-38.

28. G. N. Saridis, et al., "Stochastic approximation algorithms for system identification, estimation, and decomposition of mixtures," Proc. 5th Allerton Conf. on Circuit and System Theory, 1967, pp. 374-383; also IEEE Trans. Syst. Sci. and Cybernetics, 5, Jan 1969, pp. 8-15.

29. Y. C. Ho and R. C. K. Lee, "Identification of linear dynamic systems," Information and Control, 8, Feb 1965, pp. 93-110.

30. J. P. Comer, "Some stochastic approximation procedures for use in process control," Ann. Math. Stat., 35, 1964, pp. 1136-1145.

31. H. Kesten, "Accelerated stochastic approximation," Ann. Math. Stat., 29, 1958, pp. 41-59.

32. V. Fabian, "Stochastic approximation of minima with improved asymptotic speed," Ann. Math. Stat., 38, 1967, pp. 191-200.

33. J. Kiefer and J. Wolfowitz, "Stochastic estimation of the maximum of a regression function," Ann. Math. Stat., 23, 1952, pp. 462-466.

34. K. L. Chung, "On stochastic approximation methods," Ann. Math. Stat., 25, 1954, pp. 463-483.

35. A. Dvoretzky, "On stochastic approximation," Proc. 3rd Berkeley Symp. on Math. Stat. and Prob., J. Neyman (ed.), vol. 1, University of California Press, Berkeley, Calif., 1956.

36. K. S. Fu, Sequential Methods in Pattern Recognition and Machine Learning, Academic Press, New York, 1968 (Ch. 8)

37. A. E. Albert and L. A. Gardner, Jr., Stochastic Approximation and Nonlinear Regression, MIT Press, Cambridge, Mass., 1967.

38. M. J. Wasan, Stochastic Approximation, Cambridge University Press, 1969.

39. V. Fabian, "Stochastic approximation of constrained minima," Trans. 4th Prague Conf. Information Theory, Statistical Decision Functions, Random Processes, Academia Publishing House of the Czechoslovak Academy of Sciences, Prague, 1967, pp. 277-290.

40. R. J. Lacoss, "Adaptive combining of wideband array data for optimal reception," IEEE Trans. Geoscience Elec., 6, May, 1968, pp. 78-86.

41. B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," 1960 WESCON Conv. Rec., Institute Radio Engnrs., Part 4, pp. 96-104.

42. B. Widrow, "Adaptive filters I: Fundamentals," SEL-66-126 (TR No. 6764-6), Stanford Electronics Laboratories, Stanford, Calif., Dec 1966.

43. B. Widrow, et al., "Adaptive antenna systems," Proc. IEEE, 55, 12, Dec 1967, pp. 2143-2159.

44. L. J. Griffiths, "Signal extraction using real-time adaptation of a linear multichannel filter," SEL-68-017 (TR No. 6788-1), Stanford Electronics Laboratories, Stanford, Calif., Feb 1968.

45. L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," Proc. IEEE, 57, Oct 1969, pp. 1696-1704.

46. K. D. Senne, "Adaptive linear discrete-time estimation," SEL-68-090 (TR No. 6778-5), Stanford Electronics Laboratories, Stanford, Calif., Jun 1968.

47. J. L. Moschner, "Adaptive filtering with clipped input data," SEL-70-053 (TR No. 6796-1), Stanford Electronics Laboratories, Stanford, Calif., Aug 1970.

48. O. L. Frost, III, "Adaptive least squares optimization subject to linear equality constraints," Ph.D. dissertation, Stanford Electronics Laboratories, Stanford, Calif., Aug 1970.

49. B. Widrow, "Adaptive sampled-data systems," Proc. 1st Intl. Cong. Intl. Fed. of Automatic Cont., Moscow, 1960.

50. T. P. Daniell, "Stochastic approximation procedures for engineering applications," Proc. IEEE (letters), 57, Apr 1969, pp. 733-734.

51. T. P. Daniell, "An adaptive design procedure for environments which evolve in an unknown fashion," Proc. Conf. on Syst. Sci. and Cyber., Oct. 22-24, 1969.

52. T. P. Daniell and J. E. Brown, III, "Adaptation in nonstationary applications," to be published.

53. T. Kailath, "Sampling models for linear time-variant filters," MIT Research Laboratory of Electronics, Cambridge, Mass., TR No. 352, May, 1959.

54. H. L. Van Trees, "Analog communication over randomly-time-varying channels," IEEE Trans. Information Theory, 12, Jan 1966, pp. 51-63.

55. R. E. Kalman, "A new approach to linear filtering and prediction problems," Trans. ASME, J. Basic Eng., 82, Mar 1960, pp. 35-45.

56. R. E. Kalman and R. S. Bucy, "New results in linear filtering and prediction theory," Trans. ASME, J. Basic Eng., 83, Mar 1961, pp. 95-108.

57. H. W. Sorenson, "Kalman filtering techniques," Advances in Control Systems, vol. 3, C. T. Leondes (ed.), Academic Press, New York, 1966, pp. 219-293.

58. R. G. Bartle, The Elements of Real Analysis, John Wiley and Sons, Inc., New York, 1964.

59. H. L. Royden, Real Analysis, Macmillan Co., New York, 1968, 2nd ed.

60. D. P. Cox and H. D. Miller, The Theory of Stochastic Processes, John Wiley and Sons, Inc., New York, 1965.

61. K. L. Chung, A Course in Probability Theory, Harcourt, Brace, and World, Inc., New York, 1968.